

Nástroj pro zpracování a prezentaci statistik prodejů

Tool for Processing and Presentation of Car Sales Statistics

Vojtěch Mach

Bakalářská práce

Vedoucí: Ing. Radoslav Fasuga, Ph.D.

Ostrava, 2021

Abstrakt

Tato bakalářská práce mapuje aktuální možnosti získávání informací o automobilech a prodeji automobilů s ohledem na jejich další zpracování pro vizualizaci v podobě grafů a interaktivních tabulek. Jejím hlavním výstupem je automatizovaný nástroj pro získávání dat a znovupoužitelná komponenta pro vizualizaci těchto dat. Postupně je čtenář seznámen s aktuálními zdroji těchto dat a možnostmi jejich vizualizace pomocí rozšiřujících volně dostupných knihoven. Součástí práce je analýza zdrojů dat, specifikace zadání, analýza nástroje, návrh komponenty a její implementace. Na závěr je kapitola s úvahou o možném vylepšení komponenty. Nástroj a komponenta jsou implementovány pomocí jazyka PHP, JS a jejich rozšířeními.

Klíčová slova

vizualizace dat; interaktivní tabulky; import tabulek; PHP; JS; automatizovaný nástroj; znovupoužitelná komponenta

Abstract

This bachelor thesis maps actual possibilities of achieving informations about cars and car sales regard to their firther processing for visualization with graphs and interactive tables. It's main output is automatic tool to obtain data and reusable component for visualization. The reader is acquainted with the source of this data and possibility of their visualization using freely available libraries. Next part of thesis is data analysis, specifications, tool analysis, design and implementation. Finaly considering possible component improvements. The tool and component are implemented using PHP, JS, HTML, CSS and their extensions.

Key words

Data visualization; Interactiv table; Sheet import; PHP; JS; automatic tools; reusable component

Obsah

Úvod	- 5 -
1 Aktuální stav	- 6 -
1.1 Informace o prodeji automobilů a automobilech v ČR	- 6 -
1.2 Přehled portálů se statistikou prodeje automobilů	- 7 -
2 Analýza potřebných technických komponent pro vizualizaci výsledků	- 10 -
2.1 Google Charts	- 10 -
2.2 Chartist.js.....	- 14 -
2.3 Chart.js.....	- 16 -
2.4 Srovnání knihoven pro práci s grafy	- 19 -
3 Analýza datových zdrojů statistik	- 20 -
3.1 Svaz Dovozců Automobilů	- 20 -
3.2 Euro NCAP	- 23 -
4 Specifikace zadání pro vlastní implementaci	- 27 -
5 Analýza	- 28 -
5.1 Analýza procesů.....	- 28 -
5.2 Datová analýza	- 36 -
5.3 Stavová analýza importu a dotazování se dat pro účel vizualizace	- 43 -
5.4 Funkční analýza	- 47 -
5.5 Návrh drátěného modelu	- 50 -
6 Návrh implementace	- 52 -
6.1 Návrh technologií	- 52 -
6.2 Shrnutí návrhu implementace.....	- 54 -
7 Implementace.....	- 55 -
7.1 Struktura Programu	- 55 -
7.2 Důležité algoritmy v podobě zdrojových kódů.....	- 56 -
8 Testování	- 60 -
9 Návrh budoucího rozšíření a inovace	- 61 -
10 Závěr.....	- 62 -
Použitá literatura.....	- 63 -
Seznam příloh.....	- 65 -

Seznam použitých zkratek

Zkratka	Význam
API	Application Programming Interface
EuroNCAP	The European New Car Assessment Programme
ČSFR	Česká a Slovenská Federativní Republika
CSS	Cascading Style Sheets
JSON	Javascript Object Notation
PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
SVG	Scalable Vector Graphics
VML	Vector Markup Language
JS	JavaScript
SDA	Sdružení Dovozců Automobilů

Úvod

Cílem této práce je analyzovat možnosti získávání dat o prodeji automobilů v ČR a na jejich základě navrhnout a implementovat jejich automatizované získávání a vizualizaci.

V teoretické části jsou představeny vybrané společnosti na jejichž webových stránkách uživatel nalezne informace o automobilech. Dále jsou zde zanalyzovány vybrané knihovny pro tvorbu grafů na webových stránkách. Knihovny jsou srovnány v přehledné tabulce. Následná analýza zdrojů dat z portálů porta.sda-cia.cz a euroncap.com popisuje jejich strukturu a přínos těchto informací pro další zpracování. Analýza poskytuje potřebnou oporu pro návrh a implementaci výsledného programu. Specifikace zadání vlastní implementace a analýza procesů pomáhají při určení rozsahu implementace výsledného programu a postupu jeho algoritmů.

Výsledný program na automatizované získávání dat přijímá na vstupu tabulku ve formátu .xls z portálu portal.sda-cia.cz, či odpověď API portálu euroncap.com a jejím výstupem je uložení dat do relační databáze MySQL. Program nalezne data, jež se mají získat pomocí předdefinovaného nastavení limitů a počátků na základě analýzy zdroje dat. Proces získávání dat z portálu portal.sda-cia.cz bude implementován pomocí rozhraní pro jednoduchou záměnu procesu vkládání dat do databáze. Součástí výstupu je informace o stavu uložených dat a nových objektů v databázi. Druhá část výsledného programu je vizualizace dat v podobě interaktivních tabulek a grafů. Program bude teoreticky připraven na jednoduché zprostsředkování přístupového bodu pro získávání dat a jeho vizualizaci. V programu jsou připraveny třídy přímo určené pro získávání dat, vykreslování grafů a interaktivních tabulek.

1 Aktuální stav

1.1 Informace o prodeji automobilů a automobilech v ČR

V této kapitole je shrnutí vybraných společností, či platforem se zaměřením na informace o automobilech. Každá společnost je zde představena. Uživateli je popsána funkce webu a jeho přínos informací pro uživatele. U každé společnosti je zmíněná důvěryhodnost informací. Tyto platformy jsou zde představeny pro seznámení se s dostupnými zdroji informací o automobilech pro běžné uživatele webů.

Cebia

Společnost Cebia, zabývající se pojištěním a ochraně spotřebitelů automobilových vozidel, vlastní největší databázi záznamů ojetých vozidel ve střední Evropě. Díky tomuto velkému vzorku dat vydává každý měsíc dokument, ve kterém zobrazuje grafy o počtu prodaných ojetých automobilů, původu vozidel, udávaném stavu kilometrů, prodejní ceně, nejprodávanějších modelech, preference jednotlivých typů pohonů, vozidel s STK a bez STK, nejoblíbenějších barvách, upřednostňované výbavy, prvním majiteli. Všechny grafy až na počet prodávaných ojetých automobilů jsou v procentech, jejich přesnou hodnotu čtenář nezjistí. Daná data jsou vzhledem k renomé společnosti věrohodná, avšak jejich rozsah je značně omezený.

SDA

Navázáním na stejnojmenné sdružení ČSFR vznikl 1.4.1994 Svaz Dovozců Automobilů. Svaz je dobrovolné zájmové sdružení právnických osob ve smyslu Občanského zákoníku §20, písm. f). [1] Jejich hlavním prostředkem pro komunikaci s veřejností je webová stránka portal.sda-cia.cz, kde se nachází jejich články a články jejich partnerů, právní informace pro dovozce automobilů, seznam značek automobilů a motocyklů, kontakty, přihlášení pro členy a statistiky. Portál se skládá z dovozců automobilů nejprodávanějších značek v České Republice a střední Evropě. Jejich databáze vychází z databáze jednotlivých členů sdružení, kteří jsou jediní dovozci aut do České Republiky. Data jsou věrohodná, jelikož pochází přímo od dovozců. Dochází zde jen výjimečně k nesrovnalostem v názvech, což autor přisuzuje lidské chybě. V další kapitole jsou důkladně zanalyzována.

Autohled.cz

Webový magazín o automobilech, který uživateli jejich stránek nabízí články o jednotlivých modelech automobilů. Mimo důkladná testování automobilů také mapuje aktuální cenu na trhu v České Republice. Uživateli je poskytnuto přehledné srovnání cen jednotlivých motorizací modelu, jejich parametrů, ale také například srovnání cen sad pneumatik pro daný automobil. Web je skutečně bohatý, co se informací o jednotlivých motorizacích týče. Pro všechny modely zobrazuje dostupné motorizace včetně všech parametrů. Nabízí tak uživateli důmyslný přehled pro daný model na jedné stránce. Obsahuje také statistiku prodejů, buď po měsících, nebo celkově po barvách za určité období. Toto období si uživatel nemůže měnit. Statistiky barev vycházejí z dat MDČR, což působí důvěryhodně. Portál k modelu přikládá vlastní fotografie, či fotografie výrobce. Na závěru přehledu najdeme recenze z vybraných portálů a konkurenční modely označené jako „Rivalové“. Po zvolení konkrétní motorizace portál zobrazí průměrnou cenu povinného pojištění, havarijního pojištění, paliva ze rok, letních pneumatik, zimních pneumatik a servisu. Jednotlivé motorizace

modelů se zde dají přehledně srovnat, porovnávač jednotlivé parametry řadí vedle sebe. Uživatel tak ihned pozná rozdíly. Web slouží také jako zprostředkovatel obchodu, proto se na úvodní stránce nachází odkaz na sekci „nové vozy“ kde nalezneme dostupné automobily. Náhodnou kontrolou vzorku 20ti motorizací různých modelů nebyla zjištěna chyba v přehledu parametrů, proto je autor této práce považuje za věrohodné.

Carismo

Platforma Carismo agreguje nabídky nových a zánovních vozů od oficiálních prodejců automobilů v České Republice. Jejich předností je konfigurátor a porovnávač automobilů. Porovnávač mapuje objektivně a v přehledné formě motorizace modelu. Platforma je co se týče parametrů motorizací a výbavy vozů bohatá a přehledná, nechybí zde například informace o podpoře formátů rádia, konektivitě, či asistenčních službách. Po vybrání konkrétního modelu pomáhá s výběrem nabídky pomocí přehledného výpisu a možnosti filtrování. Konfigurátor je uživateli nápomocen, pokud shání konkrétní vůz, či má určitou představu o voze. Uživatel je systémem naveden k jeho specifikaci, výsledkem je výpis aktuálních nabídek vozů s možností odeslání poptávky prodejci. Carismo je především prodejce, proto už od úvodní stránky vybízí uživatele k nákupu, či poptávce vozidla. K zaujmutí uživatele používá také své recenze, kde zdá se vsadil na vzhled spíše než obsah. Informací je dostatek, avšak v porovnání s konkurencí ne tak obsáhlé. Data jsou od jednotlivých prodejců automobilů, platforma je pouze sdružuje, jsou tedy věrohodná.

Gloffer

Platforma Gloffer usnadňuje uživateli výběr automobilu. Stejně jako ostatní platformy zaměřené na konfiguraci, porovnání a poptávku především nových automobilů portál sdružuje informace o konfiguraci vozidel a jejich přibližné ceně v České Republice. Portál umožňuje uživateli vybrat konkrétní konfiguraci a zaslat poptávku partnerům, kteří odpovídají na poptávky uživatelů. Oproti ostatním platformám nabízí kategorizaci vozidel podle využití (rodič na plný úvazek, městský typ, dobrodruh atd.) Oproti ostatním platformám se zdá být vzhledově čistější – jednodušší, což působí přehledněji. Portál obsahuje pouze některé partnerské značky a modely, neobsahuje proto velké množství čerpatelných informací. Portál obsahuje paušálně zpoplatněný systém pro registraci prodejců automobilů. Informace tak jsou na portále v režii konkrétního prodejce, odpovídají skutečné nabídce prodejce. Informace jsou věrohodné, jelikož pochází od prodejců.

1.2 Přehled portálů se statistikou prodeje automobilů

V této kapitole jsou aktuálně dostupné portály, na kterých lze vidět statistiky prodeje automobilů podrobněji než v předešlé kapitole [1.1]. Nebude zde rozebrán jediný skutečně kvalitní a obsáhlý portál pro statistiky prodeje v České Republice – portál sda-cia.cz, jelikož je představen v předchozí kapitole [1.1] a je analyzován v kapitole [2.1]. Tyto platformy jsou odděleny od kapitoly [1.1], jelikož jsou obsáhlejší a přináší podrobnější informace podobou tabulek, či grafů o prodeji automobilů. Je třeba představit i tyto platformy, jelikož bylo zváženo jejich použití jako zdroj dat.

marklines.com

Služba poskytující svým členům informace, které jim šetří peníze a čas s ohledem na přínos informací o produkci automobilů v zemích po celém světě. [2] Tato placená služba poskytuje svým členům informace o prodeji a produkci automobilů dle zemí a kategorií, předpověď trhu

Aktuální stav

automobilů, představení nových modelů a dodavatelů. Webová stránka disponuje dnes již starším vzhledem. Vzhled je však čistě účelový a uživatel, kterého zajímají pouze informace – statistiky zde nalezne potřebné v přehledné tabulce viz. [Obrázek 1.1].

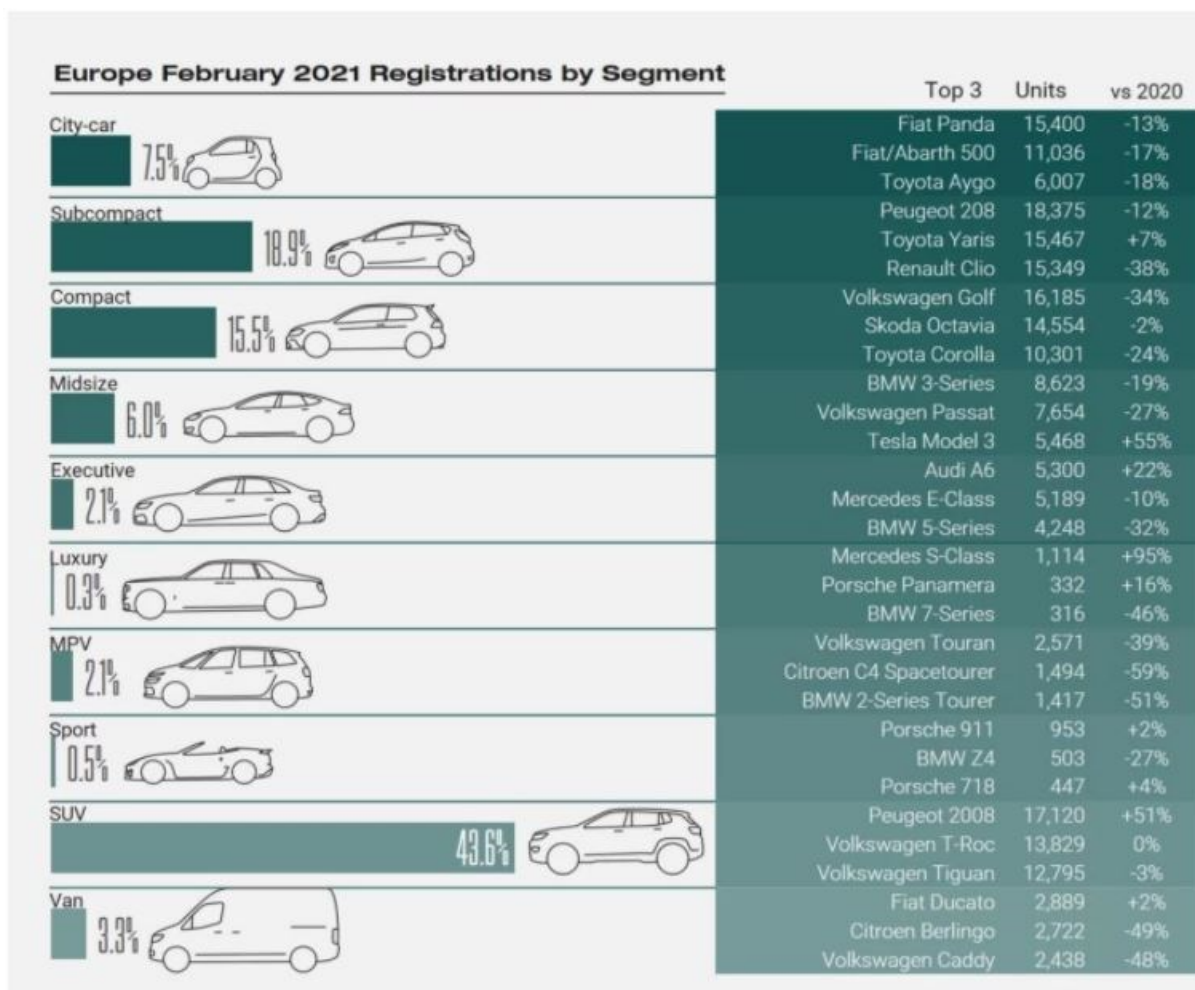
Group	Maker/Brand	Type	Segment	Model	Jan.	Feb.	Mar.	Apr.	May	Jun.	First Half (Jan.-Jun.)	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.	Second Half (Jul.-Dec.)	Adjusted value	Total	
Graph	Graph			Graph																	
VW Group	Skoda	Cars	B	Fabia	6,071	6,740	-	-	-	-	12,811	-	-	-	-	-	-	-	-	12,811	
				B Total	6,071	6,740	-	-	-	-	12,811	-	-	-	-	-	-	-	-	12,811	
			C	Octavia (Laura)	12,895	13,184	-	-	-	-	26,079	-	-	-	-	-	-	-	-	-	26,079
				Skoda SCALA	2,398	2,513	-	-	-	-	4,911	-	-	-	-	-	-	-	-	-	4,911
				C Total	15,293	15,697	-	-	-	-	30,990	-	-	-	-	-	-	-	-	-	30,990
			D	Skoda Superb	4,467	3,409	-	-	-	-	7,876	-	-	-	-	-	-	-	-	-	7,876
				D Total	4,467	3,409	-	-	-	-	7,876	-	-	-	-	-	-	-	-	-	7,876
			SUV-C	Kamiiq	5,260	6,704	-	-	-	-	11,964	-	-	-	-	-	-	-	-	-	11,964
				Karoq	6,522	6,436	-	-	-	-	12,958	-	-	-	-	-	-	-	-	-	12,958
				Kodiaq	6,181	5,777	-	-	-	-	11,958	-	-	-	-	-	-	-	-	-	11,958
				SUV-C Total	17,963	18,917	-	-	-	-	36,880	-	-	-	-	-	-	-	-	-	36,880
			Cars Total		43,794	44,763	-	-	-	-	88,557	-	-	-	-	-	-	-	-	-	88,557
		Skoda Total		43,794	44,763	-	-	-	-	88,557	-	-	-	-	-	-	-	-	-	88,557	
	VW Group Total		43,794	44,763	-	-	-	-	88,557	-	-	-	-	-	-	-	-	-	88,557		
Grand Total					43,794	44,763	-	-	-	-	88,557	-	-	-	-	-	-	-	88,557		

Obrázek 1.1: Ukázka tabulek z portálu marklines.com

Ačkoliv je web přehledně řazen do sekcí dle skupin, značek a modelů oproti statistikám z portálu SDA-CIA nedosahuje tak podrobného členění (např. paliva). Statistiky se týkají pouze prodejů nových vozidel, či jejich produkci. Statistiku ojetých automobilů zde nenalezneme.

best-selling-cars.com

Jediným dalším nalezeným řešením sdružující statistiky o prodeji automobilů je web best-selling-cars.com. Web vydává kvartálně články, ve kterých se nachází informace o prodeji automobilů členěných dle zemí. Články nevycházejí pravidelně (některé kvartály, či celé roky statistik chybí, jindy jsou zase po měsících) a jsou v různé formě (někdy jsou formou tabulky, jindy zase sloupcovým grafem), jejich počet a forma je tedy nekonzistentní. Počet zemí pro které jsou informace o prodeji dostupné je velmi omezený. Statistiky se dělí na kontinenty, či vybrané státy. Celý web působí velmi chaotickým dojmem, jelikož se zde statistiky nacházejí v nepravidelných intervalech, či nekompletní. Na [Obrázek 1.2] lze vidět graf z článku vydaného v březnu 2021 informujícím o prodeji z minulého měsíce.[3]



Obrázek 1.2: Sloupcový graf z článku na webu best-selling-cars.com

Zhodnocení

Aktuální řešení zaměřená na informace o prodeji automobilů jsou vytvořena pro potřeby analytiků, kteří tabulkám rozumí a dále s nimi pracují, či ve formě článků informují uživatele o vybraných zajímavých datech. Bezplatná vizualizace statistik prodeje se nepodařila nalézt. Všechna nalezená řešení nejsou určena běžnému uživateli se zájmem o konkrétní vůz.

2 Analýza potřebných technických komponent pro vizualizaci výsledků

V této kapitole jsou vybrané možnosti zobrazování dat v grafické podobě pomocí JavaScriptu. V kapitole se nachází výběr programových knihoven [5], které jsou hojně užívány v různých webových aplikacích. Každá knihovna je představena a je ukázáno použití nejjednoduššího grafu - sloupcového. V poslední podkapitole jsou všechny knihovny porovnány v přehledné tabulce. Jsou zde vybrány tři knihovny, které byly důkladně prostudovány pro jejich možné použití pro potřeby vizualizace dat.

2.1 Google Charts

Všem dobře známá mezinárodní společnost Google již vydala velké množství programových knihoven pro různé potřeby aplikací. Mezi ně patří také knihovna Google Charts, která se zabývá grafickým zobrazováním dat. Společnost dodává své knihovny vždy řádně zdokumentované a jinak tomu není ani u Google Charts. Jejich velká výhoda je tedy v dostupnosti a kvalitě zpracované dokumentace, čemuž napovídá už příkladů použití jednotlivých typů grafů. Jejich další pro společnost typickou výhodou je možnost škálovatelnosti dodávaného kódu k fantazii programátora.[4]

Typy grafů

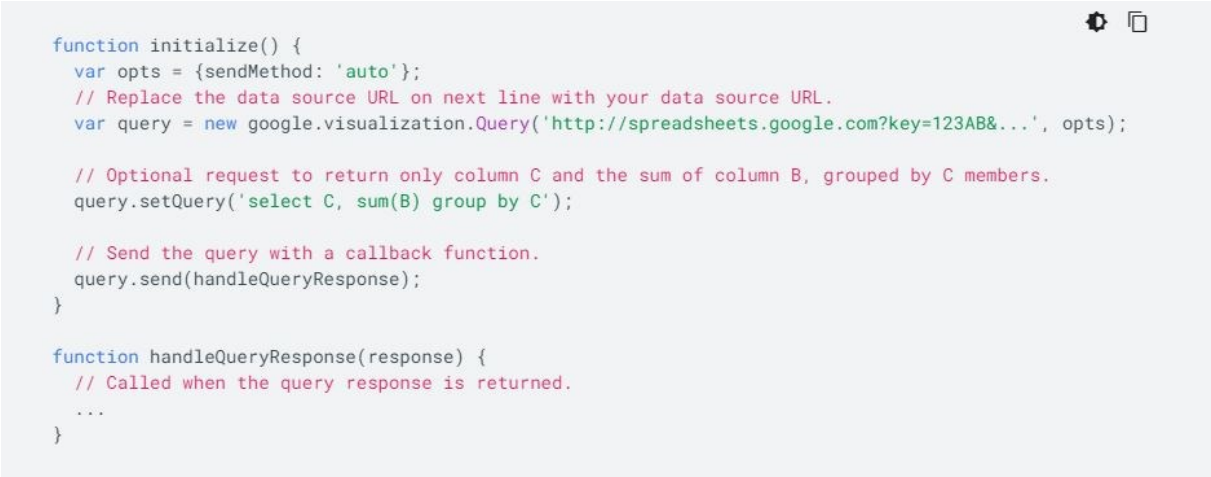
V dokumentaci Google Charts se nachází 30 druhů grafů. Od těch nejpoužívanějších, jako je sloupcový, až po méně dostupné geografy. Mezi nejzajímavější patří beze sporu geografy, kdy se programátorovi nabízí komponenta, která intuitivně pracuje s mapou a to od autora nejrozsáhlejší a nejmodernější mapy světa Google. Typy grafů jsou v knihovně reprezentovány jako JavaScriptové třídy. Ke všem typům grafů se v dokumentaci nachází alespoň pět příkladů použití. Grafy mají předdefinované základní události myši, programátor nemusí tak udělat mnoho a získá plně funkční, odladěné a interaktivní grafy. To vše je dostupné v lokalizaci automaticky nastavené dle webového prohlížeče klienta.

Formát grafu

Dokumentace se pyšní formátem kompatibilním napříč webovými prohlížeči a mobilních platforem, včetně starší verzí Internet Explorer (VML pro verze IE5 až IE9). To především díky formátu HTML5 - SVG, který zobrazuje vektorovou grafiku. Je tedy jasné, že vyobrazení budou vždy responzivní a uživatel tak dostane kvalitní obrázek grafu. Programátor nepotřebuje žádné jiné knihovny k fungování grafu.

Práce s daty

Pro práci s daty knihovna využívá třídy DataTable, která funguje na principu tabulky. Programátor do ní vkládá data pomocí konstruktoru přes JSON objekt, či postupně pomocí metod „addRows“ reprezentující řádky a „addColumn“ reprezentující sloupec. Kromě výše zmíněné třídy se pro práci s daty využijí webové služby podporující protokol Chart Tools Datasource. Ukázka práce s knihovnou [2.1]

The image shows a code editor window with a light blue background. It contains two JavaScript functions. The first function, 'initialize()', sets up a query to Google Spreadsheets. It defines an options object with 'sendMethod' set to 'auto', creates a 'query' object with a specific URL, sets a query string to 'select C, sum(B) group by C', and sends the query with a 'handleQueryResponse' callback. The second function, 'handleQueryResponse()', is a placeholder for processing the data response. In the top right corner of the code editor, there are icons for settings (a gear) and a copy function (two overlapping rectangles).

```
function initialize() {  
  var opts = {sendMethod: 'auto'};  
  // Replace the data source URL on next line with your data source URL.  
  var query = new google.visualization.Query('http://spreadsheets.google.com?key=123AB&...', opts);  
  
  // Optional request to return only column C and the sum of column B, grouped by C members.  
  query.setQuery('select C, sum(B) group by C');  
  
  // Send the query with a callback function.  
  query.send(handleQueryResponse);  
}  
  
function handleQueryResponse(response) {  
  // Called when the query response is returned.  
  ...  
}
```

Obrázek 2.1: Ukázka zdrojového kódu z dokumentace Google Charts

Na prvním řádku ve funkci `initialize` programátor zvolí typ požadavku (`xhr`, `scriptInjection`, `makeRequest`, `auto`), kdy nejuniverzálnější typ `auto` sám zvolí dle parametru `trqt` v URL zdroje dat, jaký typ se má použít. Dále inicializujeme `Query` objekt s URL zdroje dat a nastavením. Užitím metody `setQuery` si pomocí dotazu podobným SQL vybereme data, která chceme vybrat. V metodě `send` můžeme třídě `Query` předat callback, který se zavolá po vrácení dat z dotazu. Další příkladnou funkcí, kterou dokumentace zmiňuje je použití callback funkce s parametrem, který vyjadřuje data odpovědi.

Úprava komponenty

Knihovna Google Charts nabízí mnoho příležitostí, jak škálovat komponenty. Programátor může do třídy vyjadřující typ grafu agregovat data a dále nspecifikovat konkrétní vizuální prvky. V tom případě se použije základní nastavení, které nabízí kvalitní a intuitivní vzhled. V případě, že programátorovi nestačuje toto nastavení, může si nastavit veškeré vizuální prvky pomocí metod třídy daného typu grafu, či již při inicializaci třídy vložit JSON objekt s parametry. Z nastavení třídy a pomocí dalších CSS vlastností, lze například vytvořit graf připomínající palubní automobilový ukazatel či interaktivní tabulku.

Gauge Example

Temperature:



Obrázek 2.2: Ukázka úpravy z Knihovny Google Charts

Na [Obrázek 2.2] lze vidět dva interaktivní prvky „Go Faster“ a „Slow down“. Data komponenty se mohou měnit nejen při inicializaci, ale také za běhu, kdy stačí nad instancí zavolat metodu `draw` a prvek se díky předdefinovanému chování animace plynule změní.

Praktická ukázka

V této kapitole bude vysvětleno jednoduché použití sloupcového grafu, v knihovně nazvaný Bar Chart. Nejprve je třeba si načíst samotnou knihovnu Google Charts. Dále, aby byl graf umístěn na webové stránce, je třeba užítí HTML tagu `div` s jedinečným id, v příkladu `chart`.

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<div id="chart"></div>
```

Po umístění tagu `div` a naimportování Google Charts, následuje samotný kód, který je umístěn mezi HTML tagy `script`. Samotný program v těle tagů má za úkol inicializovat třídu pro sloupcový graf nad výše zmíněným `div` tagem, nastavit ji a naplnit daty.

```
google.charts.load('current', {'packages': ['bar']});
google.charts.setOnLoadCallback(drawChart);
```

Na prvním řádku ukázky říkáme programu, aby si načel knihovnu Google Charts. Metoda obsahuje dva parametry. První je verze knihovny, kdy fungují i slovní vyjádření verzí např.: `current`, `upcoming`. Druhý parametr je objekt, ve kterém specifikujeme balíky. V našem případě načítáme balík `bar`. Na druhém řádku nastavíme funkci (callback), která se má spustit po načtení knihovny, `drawChart`. Na následujícím příkladu lze vidět samotnou funkci `drawChart`.

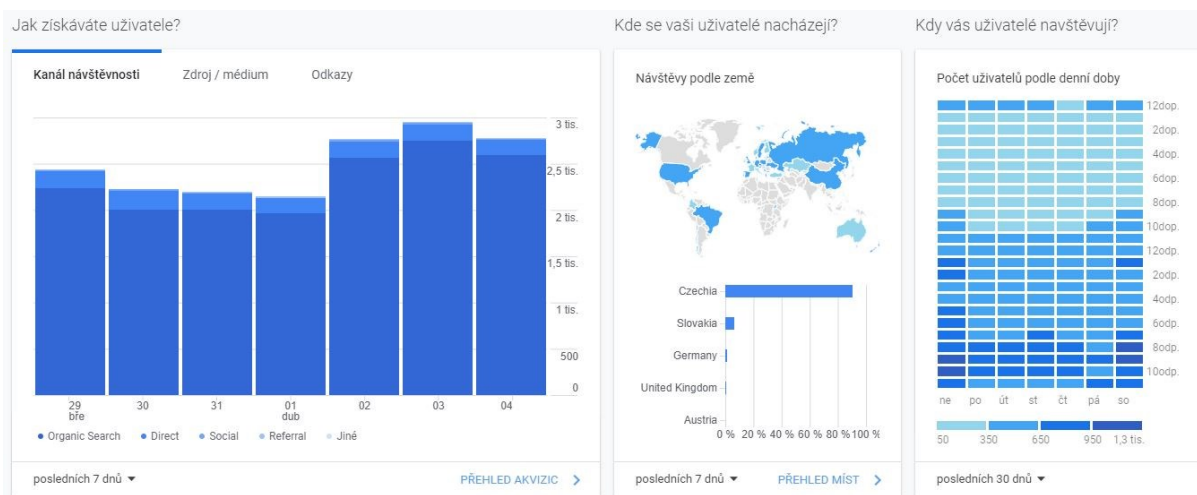
```
function drawChart() {
var data = google.visualization.arrayToDataTable([
['Rok', 'Prodej', 'Výdaje', 'Zisk'],
```

```
['2014', 100, 40, 60],  
['2015', 110, 40, 70],  
['2016', 60, 12, 48],  
['2017', 130, 54, 76]  
]);  
var options = {  
  chart: {  
    title: 'Ukázka sloupcového grafu',  
    subtitle: 'Ukázka sloupcového grafu pro bakalářskou práci',  
    height: 400,  
    colors: ['blue', 'red', 'green']  
  };  
  var chart = new google.charts.Bar(document.getElementById('chart'));  
  chart.draw(data, google.charts.Bar.convertOptions(options));  
}
```

Funkce obsahuje inicializaci proměnné data, pro kterou je v ukázce použita statická metoda z knihovny, která převede dvou, či tří dimenzionální pole na DataTable objekt. Následuje inicializace proměnné options, kdy pomocí několika málo parametrů chart : title a chart : subtitle nastavíme nadpis a podnadpis grafu. Určíme jeho výšku parametrem height a barvy sloupců parametrem colors. Do proměnné chart inicializujeme objekt Bar z knihovny Google Charts s parametrem typu objekt HTML tag div. Po inicializaci objektu Bar stačí zavolat nad objektem metodou draw překreslení s parametry data a options, přičemž pro zjednodušení zde použijeme statickou metodu knihovny Bar.convertOptions, která převede objekt options do požadovaného formátu. Pro změnu grafu, lépe řečeno dat grafu, je třeba změnit samotná data a následně nad objektem zavolat metodu draw se změněnými parametry grafu – data a options.

Zhodnocení

Knihovna jako taková obsahuje velké množství již nastavených – předpřipravených grafů, do kterých lze vložit data a programátor získá plně funkční, uživatelsky přívětivé interaktivní grafy. Oproti jiným knihovnám je obsáhlejší a vyžaduje tak větší úsilí k poznání knihovny. Díky podrobné dokumentaci má programátor veškeré informace na jednom přehledném místě rozdělených do logických celků – kapitol. To umožňuje programátorovi rychle dohledat pouze potřebné informace a nezabývat se nepotřebnými komponentami. Ačkoliv je pro použití knihovny třeba znát více, než pouze samotné grafové komponenty, hodnotí auto této práce užití pomocných objektů, jako je např. DataTable velmi kladně. Práce s nimi totiž díky dodatečné abstrakci zpřehledňuje práci s daty. Knihovna je vhodná především pro použití v přehledových panelech, jelikož disponuje již připravenými metodami pro kontrolu chování a vzhledu určených pro toto použití. Lze ji také užít pro vykreslení jednoho grafu, ačkoliv její rozsah vybízí spíše ke komplexnějšímu použití. Ukázka panelu s třemi tabulemi lze vidět na [Obrázek 2.3].



Obrázek 2.3: Ukázka přehledového panelu z Google Charts

2.2 Chartist.js

Knihovna Chartist poskytuje intuitivní použití také pro uživatele zvyklé na vytváření grafů v Excel tabulkách. Grafy jsou responzivní a ke svému fungování nepotřebují žádné další závislosti. To má za důsledek podporu různých zařízení. Jeho velká výhoda je také velikost knihovny, kdy při komprimaci gzip dosahuje pouze 10KB. [6]

Typy grafů

Knihovna se snaží být co nejjednodušší s důrazem na možnost libovolné úpravy grafů. Tomu také nasvědčuje počet předdefinovaných typů grafů. Tři typy grafů – lineární, sloupcový a koláčový a jejich možné variace nalezneme v knihovně. Další typy grafů lze implementovat úpravou těchto variací, či stažením pluginů pro knihovnu.[6]

Formát grafu

Knihovna podporuje responzivní vzhled. Díky předdefinovaným stylovým třídám lze HTML tagu místo jasně dané velikosti v pixelech určit jaký poměr stran bude výsledný graf mít. Tak je při implementaci usnadněna práce s vytvářením responzivního vzhledu. Knihovna vykresluje výsledný graf jako formát SVG v HTML tagu <div>. Formát SVG také podporuje animace svých částí pomocí CSS vlastností.

Práce s daty

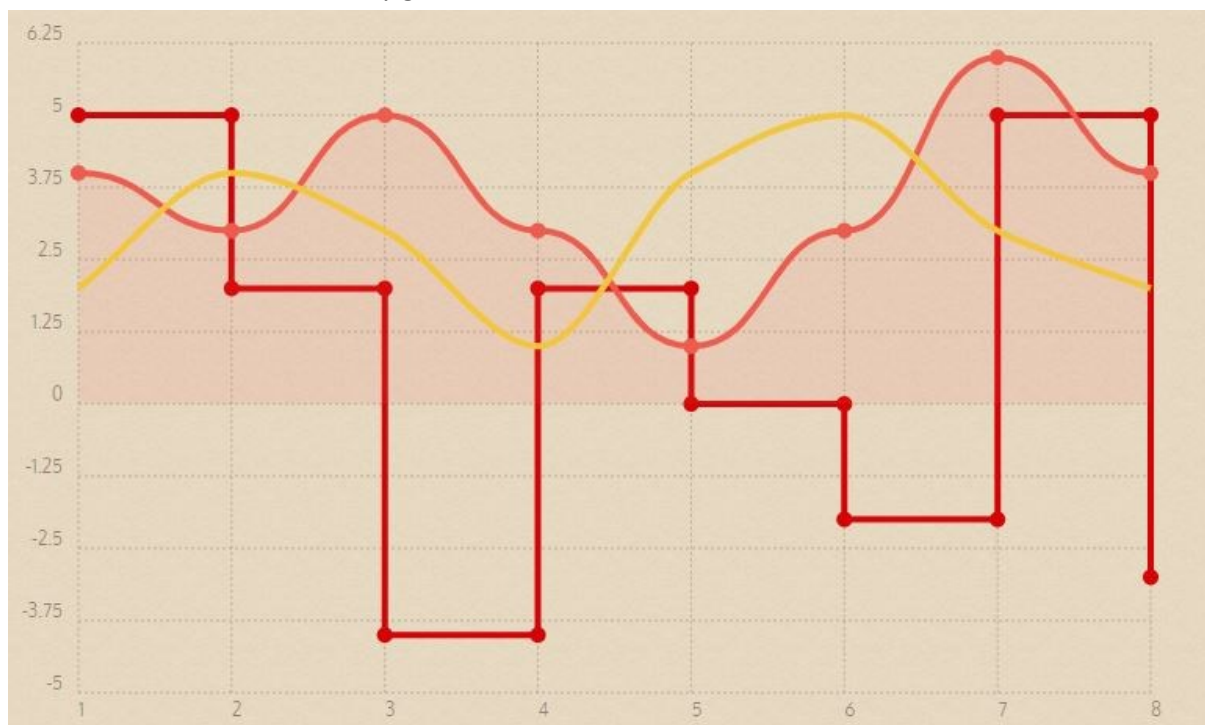
Chartist pro svá data využívá objekt typu slovník. V tomto objektu jsou data uložena pod názvem, který odpovídá povaze parametru (labels – nadpisy, series – dimenze dat). Chartist tak používá přímočarou práci s daty bez další abstrakce. Toto chování je rychle pochopitelné, intuitivní. Na ukázce zdrojového kódu níže lze vidět inicializace proměnné data s parametry labels a series potřebné k zobrazení dvou dimenzí v grafu.

```
var data = {  
  labels: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri'],  
  series: [  
    [5, 2, 4, 2, 0],  
    [1, 4, 6, 8, 1]  
  ]  
}
```

```
};
```

Úprava komponenty

Knihovna podporuje úpravu vzhledu pomocí CSS3. Podporuje také úpravu vlastností grafu pomocí SASS mixins, což umožňuje ještě větší kontrolu vlastností elementů grafu a jejich animaci. Knihovna zahrnuje řadu tříd stylů, které stačí nahradit a změnit tak vzhled výsledných grafů globálně, či pouze pro konkrétní graf. Úpravu vzhledu lze také aplikovat pouze na jednu dimenzi, graf tak může kombinovat vzhled dimenzí. Takový graf lze vidět na [Obrázek 2.4]



Obrázek 2.4: Ukázka kombinace vzhledů dimenzí v jednom grafu z dokumentace Chartist.js

Praktická ukázka

Pro praktickou ukázkou je zvolen opět sloupce graf – v knihovně Bar chart. Pro srovnání s dalšími grafy jsou použita stejná data s odpovídajícím stejným zobrazením.

```
<link rel="stylesheet" href="//cdn.jsdelivr.net/chartist.js/latest/chartist.min.css">
<script src="//cdn.jsdelivr.net/chartist.js/latest/chartist.min.js"></script>
```

Na ukázce výše lze vidět vložení knihovny do stránky. Tento fragment kódu je nutno umístit do hlavičky stránky – mezi HTML tag <head>. V dokumentaci Chartist je zmíněna jak distribuce pomocí cloudu – odkazu, tak pomocí stažení knihovny do JS aplikace pomocí správce balíčků Bower. Po vložení závislostí je třeba implementovat HTML tag <div> s identifikátorem a případně třídu pro velikost grafu. Příklad implementace tagu na ukázce zdrojového kódu níže.

```
<div class="ct-golden-section" id="chart2"></div>
```

Na ukázce je použit jako identifikátor chart2. Dále je použito třídy vlastností ct-goled-section odpovídající poměru výšky a šířky 1:1,618.

JavaScriptový kód potřebný pro zobrazení sloupcového grafu lze vidět na ukázce níže.

```
new Chartist.Bar('#chart2', {
  labels: ['2014', '2015', '2016', '2017'],
  series: [
    [100, 110, 60, 130],
    [40, 40, 12, 54],
    [60, 70, 48, 76]
  ]
});
```

Program vytvoří instanci s identifikátorem #chart2, díky čemu se graf „připojí“ k HTML tagu s tímto identifikátorem. Jako druhý parametr konstruktoru je zaslán objekt s nadpisy osy X – pole labels a dimenze dat – pole polí series.

Zhodnocení

Práce s knihovnou je velmi jednoduchá, přesně jak avizuje její dokumentace. Je velmi lehká – neobsahuje zbytečná rozšíření v balíčku. To s sebou nese určité nevýhody. Programátor musí implementovat interaktivní chování. K dosažení grafu podobným ostatním v této kapitole je tak nutno vyvinout větší úsilí. Na druhou stranu je její použití přehlednější a neobsahuje takovou míru abstrakce – je přímočará. Možnost parsování dat jsou značně omezeny na prosté naplnění pole, či definování os pomocí objektů. Při práci s knihovnou se při výskytu chyby dalo jednoduše zjistit, kde chyba nastala. Knihovnu je tak vhodné použít zejména v menších projektech s velkou škálou různých zařízení.

2.3 Chart.js

Chart.js je open-source podporující responzivní vzhled pomocí překreslování. Pyšní se vysokým výkonem v moderních prohlížečích včetně IE11+. Knihovna má pod sebou velkou komunitu aktivních uživatelů stále přispívajících do repozitáře na GitHub. Knihovna také referuje o sekci na fóru Stackoverflow s více než devíti tisíci otázkami. [7]

Typy grafů

Knihovna podporuje osm různých typů grafů [7]. Všechny jsou upravitelné a připravené k použití včetně předpřipravených animací a interaktivního chování.

Formát grafu

Knihovna k vykreslení využívá HTML5 tagu <canvas>. Ten je díky překreslování při události změny rozlišení obrazovky plně responzivní. Díky použití tagu <canvas> je také jednodušší zachytávání událostí myši a reagování na ně. Nevýhoda tohoto překreslování může nastat u pomalejších zařízení, či při animaci, kdy využívá JavaScript namísto CSS vlastností.

Práce s daty

Práce s daty v knihovně je podobná jako v předchozí kapitole. Knihovna však rozlišuje jednotlivé dimenze do datasetů. Každému datasetu lze přiřadit jak vlastnosti vzhledu (barva okrajů, barva výplně, šířka okraje atp.) tak nadpis týkající se dimenze pro potřeby interaktivních „tooltipů“. Na ukázce zdrojového kódu níže lze vidět nadpis pro nápovědu k datům. Dále lze vidět definice parametru data s jednou dimenzí její barvou výplně dimenze, barvou okrajů a šířkou okraje.

```
data: {
  labels: ['Red', 'Blue', 'Yellow', 'Green', 'Purple', 'Orange'],
  datasets: [{
    label: '# of Votes',
    data: [12, 19, 3, 5, 2, 3],
    backgroundColor: [
      'rgba(255, 99, 132, 0.2)',
      'rgba(54, 162, 235, 0.2)',
      'rgba(255, 206, 86, 0.2)',
      'rgba(75, 192, 192, 0.2)',
      'rgba(153, 102, 255, 0.2)',
      'rgba(255, 159, 64, 0.2)'
    ],
    borderColor: [
      'rgba(255, 99, 132, 1)',
      'rgba(54, 162, 235, 1)',
      'rgba(255, 206, 86, 1)',
      'rgba(75, 192, 192, 1)',
      'rgba(153, 102, 255, 1)',
      'rgba(255, 159, 64, 1)'
    ],
    borderWidth: 1
  }]
}
```

Práce s daty je tak logicky rozčleněna do pole objektů dle dimenzí. Výsledný kód je přehledný.

Úprava komponenty

Knihovna obsahuje řadu parametrů, které je možno nastavit na celý graf, či jednu dimenzi bez nutnosti úpravy externích souborů, či kaskádových stylů. Parametry obsahují buď datový typ String, Integer nebo lze přiřadit funkci - callback, který pracuje například s daty v grafu. Dokumentace pro úpravu komponenty je kvalitně zpracovaná. Při úpravách komponenty a tím také délky kódu může nastat problém při debugování úprav. Případné chyby jsou těžko dohledatelné a výsledný graf se vůbec nemusí zobrazit.

Praktická ukázka

Pro praktickou ukázkou je opět zvolen jednoduchý sloupcový graf bez specifikace vlastností vzhledu. Jsou použity stejná data s ohledem na porovnání ukázek kódu a snahou zachování zobrazení grafu.

Analýza potřebných technických komponent pro vizualizaci výsledků

Na začátku je třeba vložit knihovnu mezi HTML tag <head>. Na ukázce kódu níže je vložení knihovny.

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

Po vložení knihovny je třeba implementovat HTML tag <canvas>. Knihovna sama vytváří obalovací tag <div> pro své potřeby překreslování a pozicování. Na ukázce zdrojového kódu níže je příklad tagu <canvas> s identifikátorem chart3, pevně danou šířkou a výškou v pixelech.

```
<canvas id="chart3" width="400" height="400"></canvas>
```

Po vložení tagu následuje implementace JavaScript kódu pro inicializaci třídy Chart z knihovny Chart.js a nastavení parametrů.

```
var ctx = document.getElementById('chart3').getContext('2d');
var myChart = new Chart(ctx, {
  type: 'bar',
  data: {
    labels: ['2014', '2015', '2016', '2017'],
    datasets: [
      {
        data: [100, 110, 60, 130]
      },
      {
        data: [40, 40, 12, 54]
      },
      {
        data: [60, 70, 48, 76]
      }
    ]
  }
});
```

Na první řádce ukázky zdrojového kódu výše získá program kontext canvasu do proměnné ctx. Na druhém řádce ukázky je inicializace třídy Chart. Jako vstupní parametr konstruktoru je kontext canvasu pro vykreslení a objekt typu slovník s daty pro vykreslení. V příkladu je v objektu pro definici grafu předán type, typ grafu Bar – sloupcový graf. Dalším parametrem v objektu je data, který obsahuje nadpisy pro osu X a data dimenzí – datasets. V parametru datasets musí být alespoň jedna dimenze – objekt s parametrem data obsahující pole konkrétních čísel pro osu Y.

Zhodnocení

Komponenta je kompromisem při výběru mezi ostatními výše zmíněnými. Její použití je složitější, než v případě Chartist, avšak neobsahuje další abstrakci pro práci s daty jako v Google Charts. Její struktura je logická a snadno pochopitelná. Obsahuje velké množství předem připravených animací a interaktivního chování přizpůsobené každému typu grafů. Programátor tak nemusí implementovat

Analýza potřebných technických komponent pro vizualizaci výsledků

řadu funkcí, které jsou již v komponentě obsaženy. Ačkoliv je dle dokumentace komponenta v moderních prohlížečích výkonnější - rychlejší, než ostatní, při praktické ukázce se ukázal spíše opak. Komponenta je díky možnosti přiřadit parametrům callback snadno upravitelná. Navíc obsahuje řadu pluginů připravených pro rozšíření komponenty. Její použití se hodí především do menších až středně velkých projektů, kde jsou velkou výhodou předdefinované funkce – programátor nemusí každé chování implementovat, pouze nastavit.

2.4 Srovnání knihoven pro práci s grafy

V této podkapitole je porovnání výše zmíněných knihoven pro práci s grafy formou tabulky. V [Tabulka A.1] jsou uvedeny vybrané vlastnosti k porovnání.

Knihovna	Open-source	Počet typů grafů*	Formát grafu	Responzivní	Interaktivní chování**	Úprava vzhledu pomocí CSS	Možnost rozšíření***
Google Charts	NE	30	SVG (VML)	ANO	ANO	ANO	NE
Chartist.js	ANO	3	SVG	ANO	NE	ANO	ANO
Chart.js	ANO	8	Canvas	ANO	ANO	NE	ANO

Tabulka A.1: Srovnání knihoven pro práci s grafy

* Počet grafů knihovny bez rozšíření. Započítávají se pouze předpřipravené typy grafů.

** Zda knihovna obsahuje již připravené interaktivní chování – chování na události myši.

*** Možnost rozšíření knihovny pomocí volně dostupných pluginů.

3 Analýza datových zdrojů statistik

Tato kapitola se věnuje popisu datových zdrojů prodávaných automobilů v ČR a bezpečnostních protokolů automobilů. Zdroje jsou zde řádně zanalyzovány a následně je vyhodnoceno, která data je vhodné použít pro účely implementované komponenty s ohledem na relevanci a granularitu dat.

3.1 Svaz Dovozců Automobilů

V portálu se nachází několik dokumentů statistik, které lze zobecnit do dvou kategorií.

Statistiky zobrazené na portále SDA

Portál nabízí uživatelsky přehledné tabulky prodávaných automobilů přímo na svých webových stránkách, kde se dělí na registrace nových vozidel, registrace ojetých vozidel, vyřazená vozidla, vozový park, statistika prodeje 2002-2004, statistika prodeje 1999-2002, statistika prodeje 1993-1998. Data jsou zde zobecněna pro potřeby dovozců automobilů, přičemž nejsou vhodná pro další zpracování k potřebám této práce.

Statistiky stažitelné z portálu SDA

Na portále SDA se nachází sekce "Statistiky - Ke stažení", kde jsou každý měsíc zveřejněny čtyři dokumenty a kvartálně další dva. Tyto dokumenty jsou zde rozděleny do jednotlivých sekcí podle měsíce a roku vydání dokumentů. Každý dokument je zde pojmenován podle jasně dané šablony: "[rok]-[mesic].[typ_dat].[tabulka].[mutace].[pripona]", např.: "2020-1.mesicni.F.CZ.xls".

[rok] – Rok.

[mesic] – Měsíc.

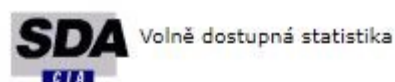
[typ_dat] – Typ dat v dokumentu (mesicni, stavvp).

[tabulka] – Pokud se jedná o tabulka, je nastaven příznak F.

[mutace] – Jazyková mutace dokumentu (CZ, EN).

[pripona] – Přípona dokumentu (pdf, xls).

Společnost Cebia, zabývající se pojištěním a ochraně spotřebitelů automobilových vozidel, vlastní největší databázi záznamů ojetých vozidel ve střední Evropě. Díky tomuto velkému vzorku dat vydává každý měsíc dokument, ve kterém zobrazuje grafy o počtu prodaných ojetých automobilů, původu vozidel, udávaném stavu kilometrů, prodejní ceně, nejprodávanějších modelech, preference jednotlivých typů pohonů, vozidel s STK a bez STK, nejoblíbenějších barvách, upřednostňované výbavy, prvním majiteli. Všechny grafy až na počet prodávaných ojetých automobilů jsou v procentech, jejich přesnou hodnotu čtenář nezjistí. Daná data jsou vzhledem k renomé společnosti věrohodná, avšak jejich rozsah je značně omezený.



leden 2021				
	2021-1.mesicni.F.CZ.pdf	Data	8,5 MB	03.02.2021 12:58:52
	2021-1.mesicni.F.EN.pdf	Data	8,5 MB	03.02.2021 12:58:52
	2021-1.mesicni.F.CZ.xls	Data	1,91 MB	03.02.2021 12:58:20
	2021-1.mesicni.F.EN.xls	Data	1,89 MB	03.02.2021 12:58:14
prosinec 2020				
	2020-12.mesicni.F.CZ.pdf	Data	9,6 MB	06.01.2021 10:32:16
	2020-12.mesicni.F.EN.pdf	Data	9,6 MB	06.01.2021 10:31:34
	2020-12.stavvp.CZ.pdf	Stav vozového parku	45,6 kB	22.01.2021 8:25:45
	2020-12.stavvp.EN.pdf	Stav vozového parku	45,0 kB	22.01.2021 8:25:45
	2020-12.mesicni.F.CZ.xls	Data	2,39 MB	06.01.2021 10:30:52
	2020-12.mesicni.F.EN.xls	Data	2,37 MB	06.01.2021 10:31:00

Obrázek 2.1: Ukázka stažitelných dokumentů na portále SDA-CIA

Dokumenty vycházející každý měsíc jsou vždy alespoň čtyři a vycházejí již od ledna 2005. Tyto čtyři dokumenty jsou téměř shodné, liší se pouze v jazykové mutaci a formátu dat. Portál SDA podporuje mimo české také anglickou mutaci, obě mutace jsou zde dostupné jak ve formátu xls, tak ve formátu pdf.

Dokumenty vycházející kvartálně se začaly používat od března 2014. Jedná se vždy o dva soubory, přičemž rozdíl je pouze v jazykové mutaci. Podporována je mimo českou pouze jedna jazyková mutace - anglická. V dokumentech se nachází aktuální stav vozového parku podle roku a kategorií jako OA - osobní automobily, LUV - lehké užitkové vozidla aj. Z dat lze vyčíst aktuální stav kusů, rozdíl oproti let předešlých a průměrné stáří vozidel.

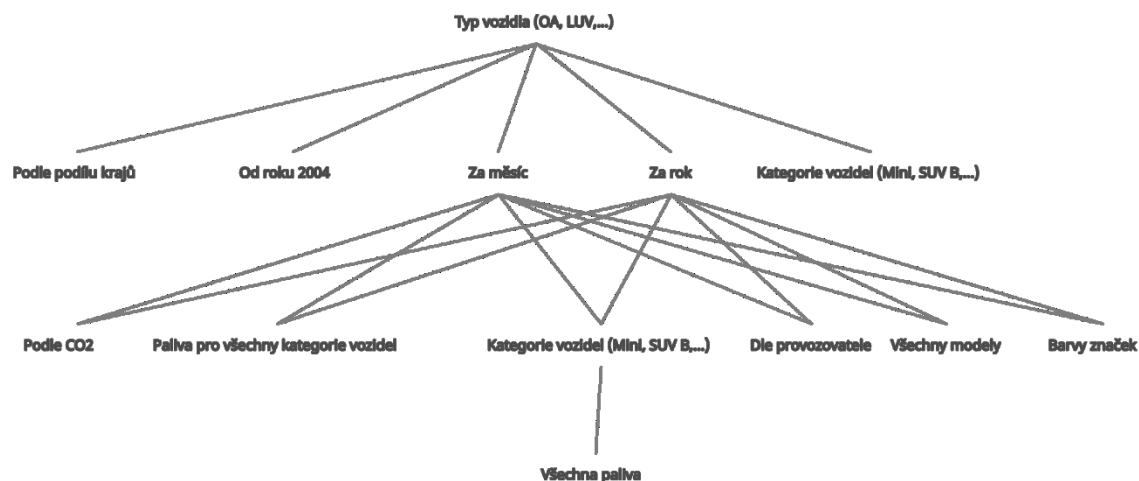
Z těchto dokumentů je nejvhodnější formát xls vzhledem k jeho charakteristikám a způsobům používání. Při zvážení budoucího užití dat a to především jeho lokalizaci - Česká republika je zvolena česká mutace dokumentu.

Měsíční dokument statistiky ve formátu xls

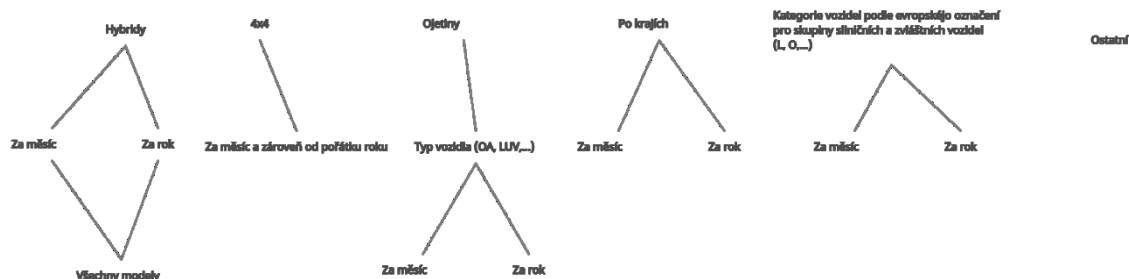
Tento soubor vychází každý měsíc zpětně k danému měsíci, bez pevně daného dne vydání v měsíci. Statisticky vychází nejpozději 28. den v měsíci.

Rozdělení tabulek na jednotlivé záložky

V dokumentu se nachází 244 záložek (aktuální k 13.2.2021), které se dle autora této práce dají rozčlenit do kategorií a podkategorií, znázorněné stromy - lesem na [Obrázek 2.2] a [Obrázek 2.3]



Obrázek 2.2: Strom záložek



Obrázek 2.3: Les záložek

V těchto záložkách se nachází počet prodaných vozů, přičemž některé záložky jsou pouze jiný pohled nad data. Obecně pro tvorbu databázového modelu je vhodné použít data s největší granularitou. Užitím tohoto pravidla v tomto případě vyplývá užití kategorií s největším zanořením, např.: Typ vozidla – Za měsíc – Kategorie vozidla – Všechna paliva. Výpis konkrétních záložek vhodných k dalšímu zpracování se nachází v [Příloha A.]

Registrace nových OA v ČR za měsíc dle paliva - Mini
1 / 2021

Značka	Palivo											Celkem	Podíl
	Benzín	Nafta	CNG	Nafta + CNG	Benzín + CNG	Benzín + LPG	E85	Elektro	Benzín + El.	Nafta + El.	Jiné		
Citroën C1	6	-	-	-	-	-	-	-	-	-	-	6	2,61%
Fiat 500	5	-	-	-	-	-	-	-	-	-	-	5	2,17%
Fiat PANDA	24	-	-	-	-	1	-	-	-	-	-	25	10,67%
Hyundai i10	41	-	-	-	-	-	-	-	-	-	-	41	17,63%
Kia PICANTO	8	-	-	-	-	-	-	-	-	-	-	8	3,48%
Mitsubishi SPACE STAR	7	-	-	-	-	-	-	-	-	-	-	7	3,04%
Peugeot 108	34	-	-	-	-	-	-	-	-	-	-	34	14,78%
Suzuki IGNIS	35	-	-	-	-	-	-	-	-	-	-	35	15,22%
Škoda CITIGO	-	-	-	-	-	-	-	20	-	-	-	20	8,70%
Toyota AYGO	46	-	-	-	-	-	-	-	-	-	-	46	20,00%
Volkswagen UP!	3	-	-	-	-	-	-	-	-	-	-	3	1,30%
Celkem													
Celkem	209	-	-	-	-	1	-	20	-	-	-	230	
Podíl	90,87	0	0	0	0	0,43	0	8,7	0	0	0	100	

Zdroj dat: Centrální registr vozidel Ministerstva dopravy ČR.
Pro SDA ve spolupráci s MD ČR, CRV a Sdružením AP zpracovala řa RTV data, s.r.o.

Obrázek 2.4: Ukázka záložky "OA Paliva za měsíc Mini"

Při pohledu na [Obrázek 2.4] je jasně vidět typická struktura tabulky. Lze si také povšimnout jednotlivých barev buněk, jsou užity barvy v hexadecimálním formátu:

- #808080
- #e6e6e6
- #bbbbbb
- #b3b3b3
- #dddddd
- #f0f9ff - (Jiné záložky)
- #ddeeff - (Jiné záložky)

Jednotlivé barvy se po dobu trvání vydávání dokumentu nezměnily.

Správnost dat

V buňkách tabulek se nachází data. Data se dají rozlišit na dva typy:

- Nadpis řádku nebo sloupce
- Konkrétní data

Při analýze tabulek a jejich zpracování bylo zjištěno několik anomálií. Tyto anomálie vznikly lidskou chybou a vyskytovaly se pouze v „Nadpis řádku nebo sloupce,“ např.: Benzin = Benzín.

3.2 Euro NCAP

Nadnárodní sdružení se zájmem o bezpečnost vozidel založené v roce 1997. Euro NCAP poskytuje nezávislé hodnocení bezpečnosti nejprodávanějších vozidel v Evropě. Na jejich stránkách www.euroncap.com vychází kromě hodnocení také články týkající se především bezpečnosti vozidel a jejich uzákoněním.

Stránka s hodnocením

Při práci se stránkou pro hodnocení automobilů užívá aplikace API na adrese www.euroncap.com/Umbraco/EuroNCAP/SearchApi/GetAssessmentSearch. K dané API neexistuje volně přístupná dokumentace, autorovi této práce se jí však podařilo částečně zanalyzovat metodou pokus - omyl. Ukázka stránky s hodnocením je na [Obrázek 2.5]

Advanced Search

Rating year ☐ All

☐ 2021 ☐ 2020 ☐ 2019 ☐ 2018 ☐ 2017 ☒ 2016 ☐ Pre-2015

Safety Equipment

☒ Standard safety equipment ☒ With safety pack

Alfa Romeo All

+ MORE FILTER OPTIONS

2016 - Rating **ABOUT 2016 RATING**

Make & Model	Safety Equipment	Overall rating	Driver Assistance Technologies	Other
Alfa Romeo Giulia	Standard	★★★★★	98%	81% 69% 60%

Obrázek 2.5: Ukázka filtrů a jeho výsledku na stránce EuroNCAP

Při zaškrtnutí (povolení) roku 2016, všech ostatních filtrů a volby značky Alfa Romeo vznikla adresa:

„<https://www.euroncap.com/Umbraco/EuroNCAP/SearchApi/GetAssessmentSearch?protocols=24370&make=7276&model=0&carClasses=1202,1199,1201,1196,1205,1203,1198,1179,40250,1197,1204,1180,34736&driverAssistanceTechnologies=&allProtocols=false&allClasses=true&allDriverAssistanceTechnologies=false&includeFullSafetyPackage=false&includeStandardSafetyPackage=true&showOnlyHybrid=false&showOnlyFleet=false&starNumber=&thirdRowFitment=false>“

Z této adresy jsou jasně čitelné parametry, které se vyplňují při každém požadavku. U parametrů bylo výše zmíněnou metodou pokus - omyl zjištěno:

protocols

- Zde patří jedno, či více ID pro daný rok od roku 2015 oddělené čárkou.
- Autorovi se nepodařilo zjistit způsob, jakým je získáváno ID pro daný rok. K jeho zjištění je třeba povolit daný rok ve filtru a nechat stránku vygenerovat dané ID.

make

- Nabývá ID pro danou značku, či 0 pro všechny značky.
- ID značky je čitelné ze zdrojového kódu stránky.

model

- Nabývá ID daného modelu, či 0 pro všechny modely.
- ID modelu je čitelné ze zdrojového kódu stránky.

carClasses

- Nabývá ID dané třídy, či více ID tříd.,

- ID třídy je čitelné ze zdrojového kódu stránky.
driverAssistanceTechnologies
- Nepodařilo se zjistit – neovlivnilo výsledek dat.
allProtocols
- Nabývá hodnoty true, pokud chceme i protokoly zvané „preAssessments“ – tedy protokoly před rokem 2015
- V opačném případě vrací prázdný objekt.
allClasses
- Nepodařilo se zjistit – nemělo vliv na požadovaný výsledek.
allDriverAssistanceTechnologies
- Nabývá hodnoty true, pokud chceme i data s konkrétními asistenčními technologiemi
includeFullSafetyPackage
- Nabývá hodnoty true, pokud chceme i data s plnou výbavou asistenčních technologií
includeStandardSafetyPackage
- Nabývá hodnoty true, pokud chceme i data s standardní výbavou asistenčních technologií
showOnlyHybrid
- Nabývá hodnoty true, pokud chceme data pouze o hybridních automobilech
showOnlyFeet
- Nepodařilo se zjistit
starNumber
- Nabývá true, pokud chceme hodnocení ve Star rating standardu dle Euro NCAP (viz. dále).
thirdFitment
- Nabývá true, pokud chceme hodnocení s třetí řadou sedadel.

Odpověď

Odpověď GET požadavku je ve formátu JSON. Odpověď na příklad požadavků výše se nachází v [Příloha B]. Odpověď se dělí vždy na dva objekty:

- AssessmentSearchResults – Hodnocení vytvořeno po roce 2015
- PreAssessmentSearchResults - Hodnocení vytvořeno před rokem 2015
- AssessmentSearchResults je pole objektů. Porovnáním objektů v poli a výsledků na stránce Euro NCAP bylo zjištěno:
 - „ProtocolId“ – ID roku protokolu.
 - “ProtocolYear” – rok protokolu.
 - “Assessments” – Pole objektů - hodnocení modelů
- Assessments objekty obsahují celou řadu pro potřeby našeho databázového modelu nepotřebných informací, a proto zde dále zanalyzují pouze ty, které je vhodné použít.
- Stars – Hodnocení pomocí Star rating.
- OverallRating – Celkové hodnocení bezpečnosti.
- ChildOccupantRating – Hodnocení bezpečnosti šestiletého a desetiletého dítěte.
- AdultOccupantRating – Hodnocení bezpečnosti dospělého spolujezdce.
- PedestrianRating – Hodnocení bezpečnosti řidiče.
- SafeAssistRating – Hodnocení bezpečnostních asistentů.

- ProtocolId – ID pro daný protokol – rok.
- MakeId – ID dané značky.
- ModelId – ID modelu.
- Make – Název značky.
- Model – Název modelu.

Parametr Stars odpovídající hodnotě standardu EuroNCAP je bodovým ohodnocením dostupnosti modelů a jejich bezpečnostní výbavě v Evropě. Parametry končící „Rating“ nabývají hodnot 0 až 100, vyjadřují procentuální úspěšnost. Dle Euro NCAP předpisů se tato hodnocení co dva roky aktualizují, kvůli zachování jednotného hodnocení po dobu času (100% v roce 2015 nemusí odpovídat 100% v roce 2020). ProtocolId, MakeId, ModelId odpovídají interním ID daného objektu. Make a Model jsou platnými názvy automobilů dle výrobců.

4 Specifikace zadání pro vlastní implementaci

V této kapitole je shrnutí zadání pro implementaci požadovaného programu. Jeho funkcionality a případná úskalí, která mohou nastat při plnění požadavků.

Pro vypracování programu je nutno, aby autor vyřešil:

- Metoda unifikace modelů, značek a skupin automobilů tak, aby se výsledné statistiky daly jednotně vizualizovat a porovnávat jejich výsledky
- Implementace získávání dat a jejich automatizace z portálu SDA-CIA
- Implementace získávání dat a jejich automatizace z portálu EuroNCAP
- Implementace vhodné struktury pro ukládání dat
- Vizualizace a porovnávání dat
- Znovupoužitelnost komponenty

Již při specifikaci prvního bodu nastává otázka, zda-li budou data ze dvou portálů (SDA-CIA a EuroNCAP) navzájem kompatibilní, či zda se podaří nalézt algoritmus pro jejich unifikaci. Z předchozí kapitoly je zřejmé, že názvy modelů a skupin nejsou shodné. Dle analýzy z předchozí kapitoly obou portálů lze také zjistit, jak lze data získávat. Oba portály pracují s jiným zdrojem dat. Data z portálu SDA-CIA mohou měnit strukturu, při návrhu a implementaci je tak nutno pracovat s každým portálem zvlášť, nalézt dostatečně komplexní řešení pro jednotlivé portály a zbytečně nelpět na obecném řešení. Při návrhu a implementaci vhodné databázové struktury je především z důvodů užití dvou různých portálů nutno myslet na fakt, že data nejsou unifikovaná. Je tedy žádoucí si uchovávat data jak v původní podobě, tak data v unifikované podobě. Pro vizualizaci dat je vhodné dbát pravidel pro vizualizaci dat. Tyto pravidla zajišťují, aby na uživatele graf působil čitelně a přehledně. Znovupoužitelnosti komponenty lze dosáhnout mnoha způsoby. Autorovi řešení bylo vedoucím práce doporučeno řešení znovupoužitelnosti, je tedy žádoucí, aby dané řešení náležitě zanalyzoval a implementoval.

Další požadavky komponenty:

- Programovací jazyk PHP nebo Javascript
- Responzivní vzhled
- Možnost vytvoření API
- Možnost nastavení barev grafu

Požadavek na programovací jazyk PHP vznikl při analýze specifikace, jelikož je hojně využíván a podporují ho dosavadní serverová řešení portálu Gloffer. Javascript již dnes podporují všechny moderní prohlížeče, není tak důvod jej nevyužít k implementaci interaktivních prvků komponenty. Responzivní vzhled podporují všechny moderní webové aplikace, je tak žádoucí, aby ji výsledný program podporoval. Při konzultaci, především z důvodů budoucího použití, či zlepšení, vznikla myšlenka vytvoření API. Výsledná architektura programu by tak s ohledem na budoucí rozšíření programu neměla klást překážky pro vytvoření přístupového bodu. Z důvodu použití komponenty v různých aplikacích, kdy každá aplikace může obsahovat jinou paletu barev, je nutno při analýze a následné implementaci zohlednit snadnou změnu barev pro grafy statistik.

5 Analýza

V této kapitole je provedena analýza. Data zdrojů již byla zanalyzována v kapitole [2.], jejich struktura a formát již zde nebudou zmíněny. Budou zde zanalyzovány procesy, více k analýze procesů zde [15]. K analýze procesů je využíváno jazyka UML, o se lze více dočíst zde[16][17]. Kapitola pouze krátce analyzuje vzhled komponenty a případného seskupení více komponent pro potřeby prezentování modelu, značky, či skupiny automobilů v drátěném modelu. O drátěném modelu lze přečíst více zde [18]

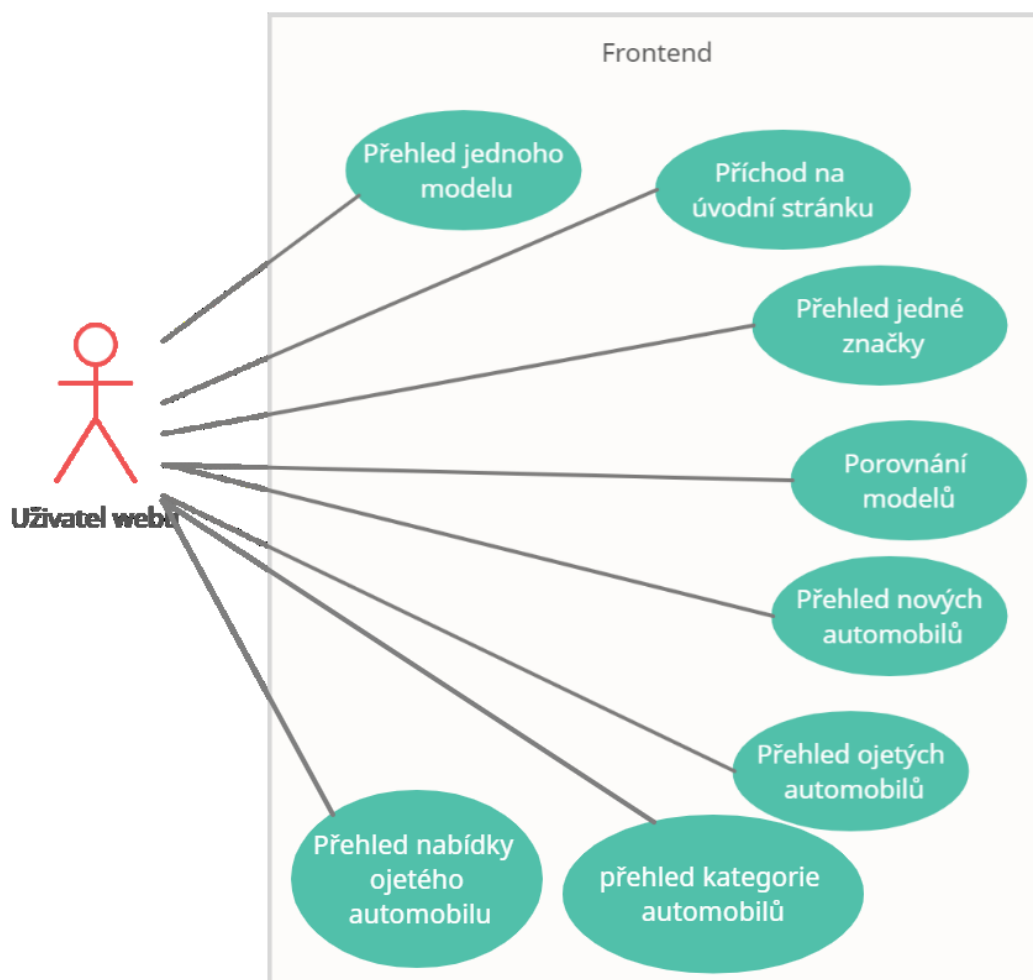
5.1 Analýza procesů

Pro navržení a implementaci komponenty je důležité popsat procesy, které budou uživatelé při práci s ní vykonávat. Kvalitní analýza procesů zjednoduší návrh rozhraní a slouží jako opora při návrhu implementace. K určení procesů, ke kterým může dojít byl použit především portál Gloffer.

Při analýze procesů, bylo zjištěno:

- Příchod uživatele na portál zaměřený na automobily
- Přehled modelů jedné značky automobilů
- Přehled konkrétního jednoho modelu, či jeho konfigurace
- Přehled kategorie automobilů
- Porovnání n-různých automobilů
- Přehled aktuálně nabízených nových, či zánovních automobilů
- Přehled aktuálně nabízených ojetých automobilů
- Přehled nabídky ojetého vozidla

Z výpisu těchto procesů vyplývá šest případů užití, které je nutno zahrnout v analýze procesů. Těchto šest zobecněných případů užití pro návštěvníka webu lze také vidět na [Obrázek 5.1].

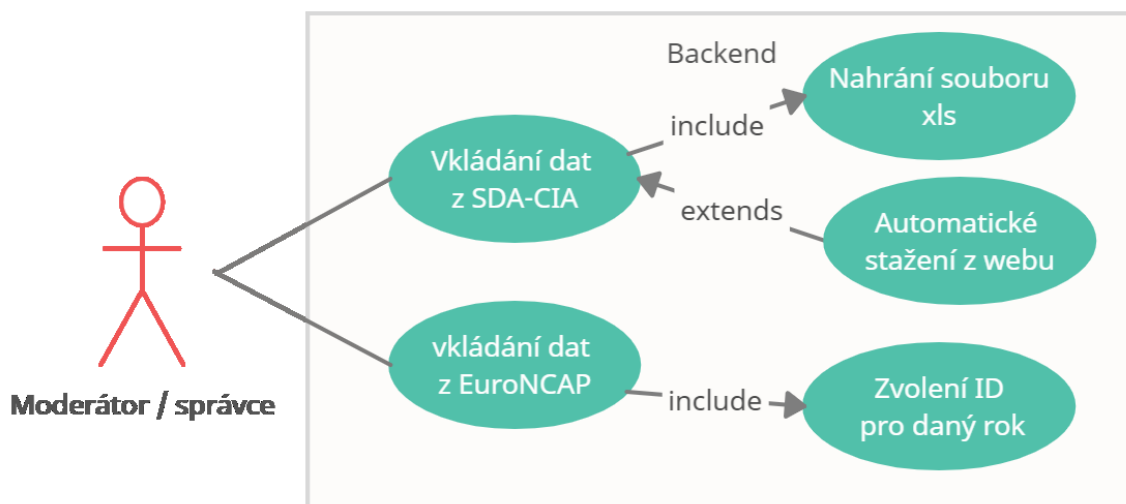


Obrázek 5.1: UML diagram - zjednodušený use case diagram z pohledu návštěvníka webu zaměřeného na automobily

Komponenta, ačkoliv má být jakkoliv automatická bude obsahovat procesy ve vztahu se správcem komponenty. Budou to tyto procesy:

- Vkládání dat z portálu SDA-CIA
- Vkládání dat z portálu EuroNCAP

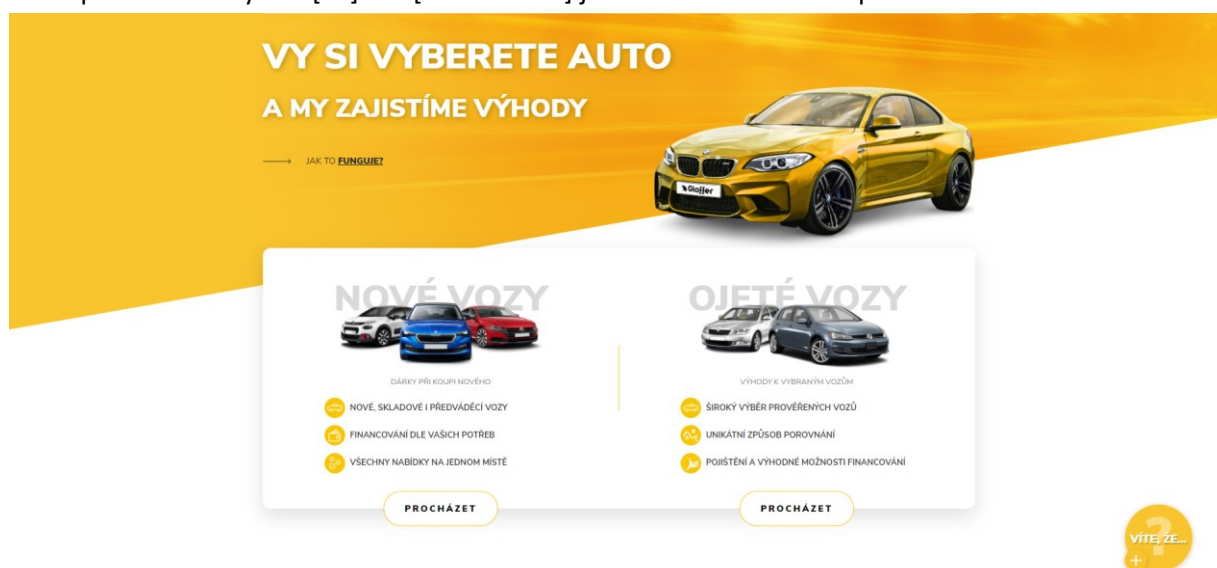
Procesy, nemusí vykonávat programátor. Může je vykonávat moderátor portálu bez praktických znalostí programování, či je implementovat programátor jako samo spustitelný kód. Interakce s uživatelem není nutná, záleží však na integraci komponenty do stávajícího řešení. Na [Obrázek 5.2] jsou zobrazeny interakce se správcem komponenty.



Obrázek 5.2: UML diagram - zjednodušený use case diagram z pohledu správy webu

Příchod uživatele na portál zaměřený na automobily

Každý úspěšný webový portál obsahuje úvodní stránku. Portály zahrnuté v kapitole [1.] také mají úvodní stránku, která příchozího uživatele postupně seznamuje se zaměřením portálu. Portály, které vydělávají na zprostředkování prodeje odkazem vybízejí k poptávce automobilu již na úvodní stránce. Účelem této stránky je uživatele nejen seznámit se zaměřením portálu, ale také jej zaujmout, více o účelu pvodní stranky zde [19]. Na [Obrázek 5.3] je vidět úvodní stránka portálu Gloffer.



Obrázek 5.3: Pohled na úvodní stránku portálu Gloffer

Úvodní stránka obsahuje tzv. infografiku, tedy graficky znázorněné informace. Více o tom, co je infografika a jak ji tvořit zde [20]. Pro dosažení snadno pochopitelné infografiky se často užívá ikon, či znázornění šipek k vyjádření směrů, předmětů a činností. Správná vizualizace dat statistiky prodeje v podobě grafů splňuje podmínky infografiky, více zde [13][14].

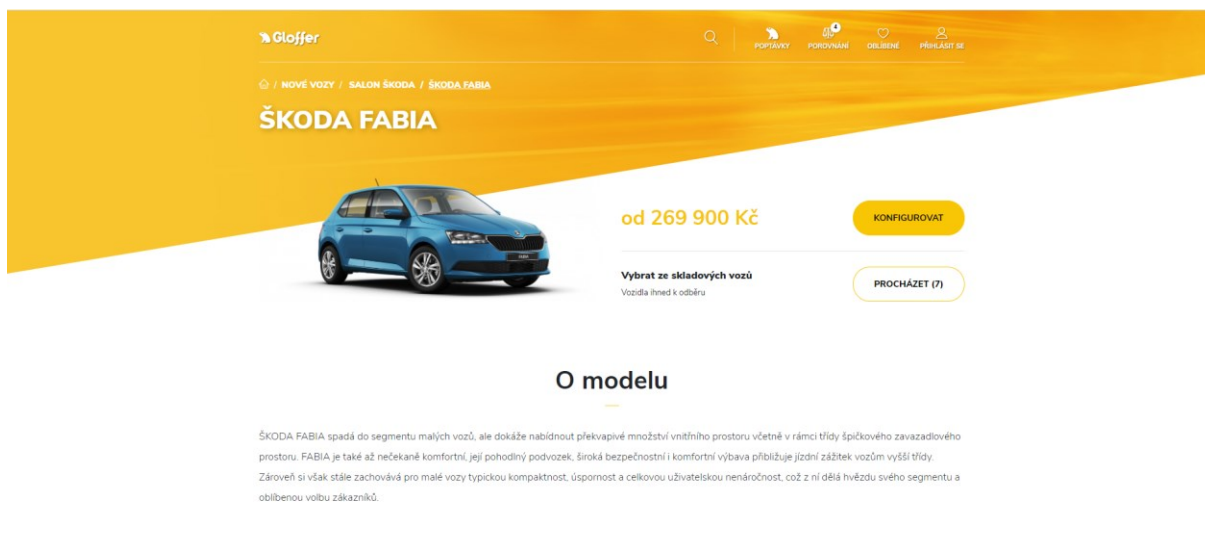
- Podat informaci přehledně a srozumitelně
- Prezentovat pouze fakta
- Být vizuálně přitažlivá

- Vizualizace dat v podobě grafů, týkajících se zájmu portálu tak může přispět k zaujmutí uživatele vzhledem, či informacemi.

Přehled konkrétního jednoho modelu, či jeho konfigurace

Uživatel navštěvující portál s automobily může zajímat konkrétní automobil. Proto je téměř nezbytné, aby obsahoval přehled konkrétního modelu automobilu. Tato stránka lze vidět na [Obrázek 5.4], typicky obsahuje informace týkající se:

- Informace o motorizaci
- Informace o výbavě
- Konkurenční modely



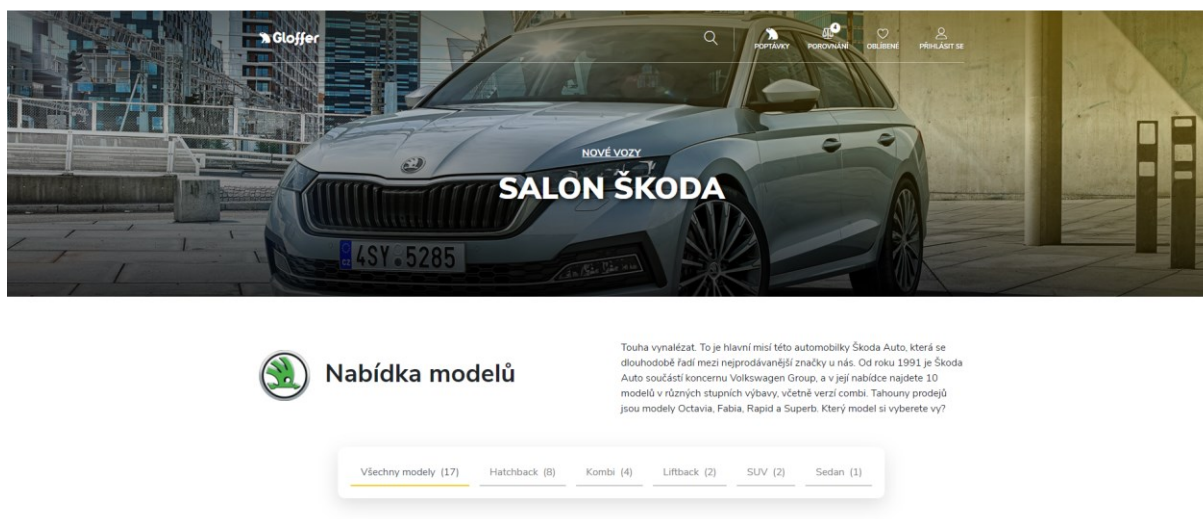
Obrázek 5.4: Pohled na stránku modelu Škoda FABIA z portálu Gloffer

Pro uživatele je přínosem vidět informace přehledně seskupené dle konkrétního zájmu. V případě modelu mu tak statistiky týkajících se konkrétního modelu a jeho konkurentech mohou poskytnout důležité srovnání. Dle dostupné analýzy dat ze zdrojů, vykonávaných procesů na portálech a dostupných informacích na portálech tak lze vytvořit seznam těchto statistik.

- Celkové porovnání modelu v nejprodávanější skupině za určité období.
- Porovnání okolí modelu v nejprodávanější skupině za určité období.
- Celkové porovnání modelu pro svou značku za určité období.
- Porovnání okolí modelu pro svou značku za určité období.
- Prodeje modelu po měsících za určité období.
- Prodeje modelu po měsících podle paliv za určité období.
- Celkové prodeje modelu podle paliv za určité období.
- Zobrazení vybraných informací z bezpečnostních protokolů modelu.

Přehled modelů jedné značky automobilů

Přehled modelů značky slouží také jako úvodní stránka značky. Má tak uživatele krátce seznámit s automobilkou. Z pohledu na [Obrázek 5.5] je patrné seznámení s automobilkou Škoda. Uživatel vidí všechny modely automobilky s obrázkem a jejich krátkým popisem.



Obrázek 5.5: Pohled na stránku značky Škoda z portálu Gloffer

Dle dostupné analýzy dat ze zdrojů, vykonávaných procesů na portálech a dostupných informací na stránkách tak lze vytvořit seznam těchto statistik.

- Celkové porovnání modelů značky za určité období.
- Prodeje modelů značky po měsících za určité období.
- Nejprodávější typy paliv automobilky.
- Nejprodávější skupiny automobilů značky.
- Prodeje modelů po měsících podle paliv za určité období.
- Prodeje skupin automobilů po měsících za určité období.

Přehled kategorie automobilů

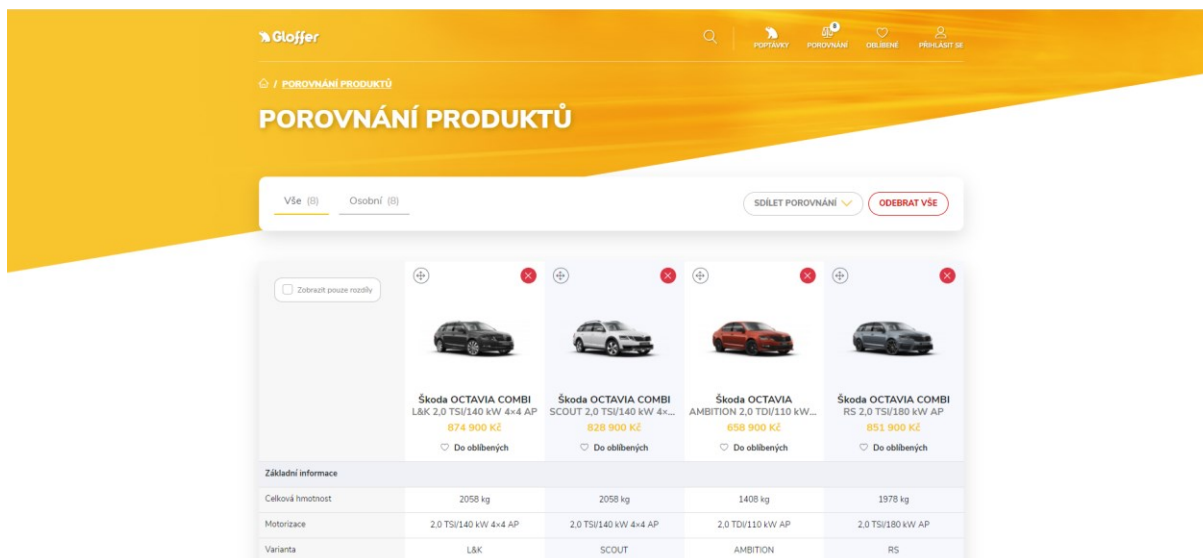
Z portálů v kapitole [1.1] žádný přímo neobsahuje rozlišení kategorií dle analýzy z kapitoly [2.1]. Přesto je zde zařazena z důvodu jejího možného použití v zobecněných kategoriích, např. na portále Gloffer, či autohled.cz. Každý z portálu obsahuje jiné kategorie, dle životního stylu, či karoserie vozidla. S ohledem na zobecněné kategorie portálů a analýzy v kapitole [2.1] lze vytvořit tento seznam statistik.

- Celková prodávánost modelů v kategorii za určité období.
- Proávánost modelů automobilů v kategorii po měsících za určité období.
- Celková prodejnost značek v kategorii za určité období.
- Proávánost značek automobilů v kategorii po měsících za určité období.
- Celkové prodeje paliv v kategorii za určité období.
- Proávánost paliv v kategorii za určité období.

Porovnání n-různých modelů

Portály z kapitoly [1.1] se od webových stránek zaměřených na prodej, či jeho zprostředkování liší především informacemi a jejich zdroji. V portálech informace dodává správce webu, který často čerpá od výrobce. Na webových stránkách zaměřených na přeprodáv ojetých automobilů vkládá uživatel informace o svém automobilu - inzerátu. Informace se tak mohou lišit jak kvalitou, tak množstvím. Srovnání několika inzerátů ojetých automobilů, především jeho kvalita a přínos pro uživatele, přímo nezávisí na správci webu. Z pohledu portálů je tak srovnání několika automobilů

jednoznačným přínosem pro uživatele. Na [Obrázek 5.6] můžeme vidět srovnání čtyř variant Škoda OCTAVIA (Laurina & Klement, Scout, RS, Style). Přehledná vizualizace dat prodejů automobilů tak může uživateli poskytnout další srovnání.



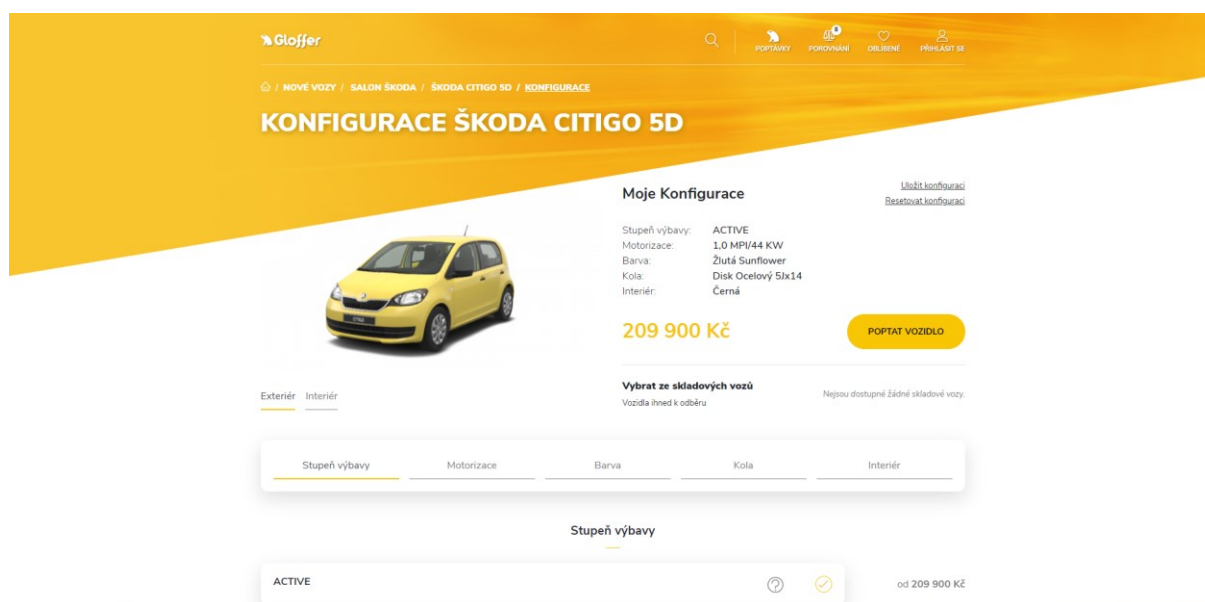
Obrázek 5.6: Pohled na srovnání čtyř různých variant modelu Škoda OCTAVIA

Při analýze zdroje dat v kapitole [2.] je patrná granularita dat statistik. V porovnání s portály, které nabízí podrobnější informace o konkrétních variantách automobilů, jsou statistiky prodejů obecné. Je tak třeba důkladně zvážit jejich použití v případě, kdy uživatel zvolí různé varianty jednoho automobilu. Vzhledem k analýze [2.], informací dostupných na portálech a možnosti dalších procesů uživatele vznikl seznam možných vizualizací dat.

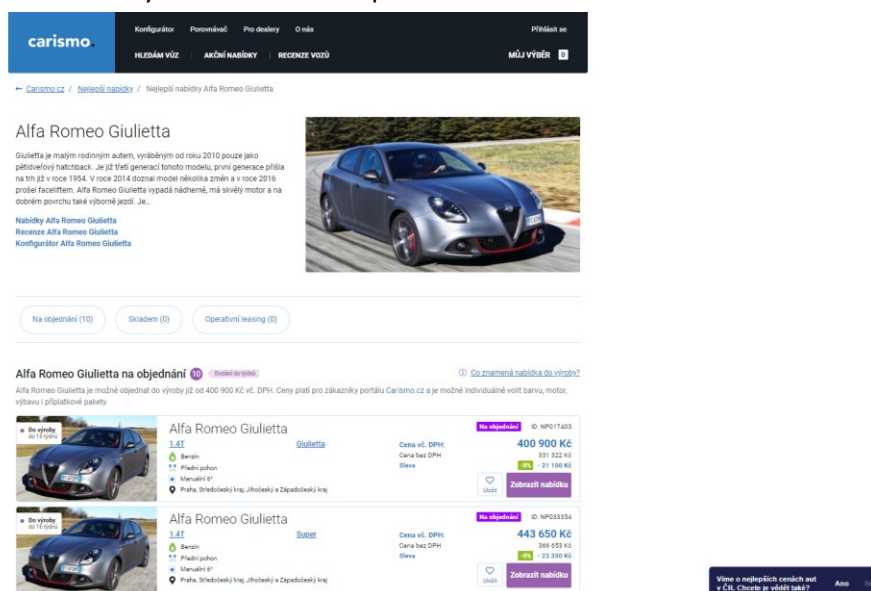
- Srovnání n-různých prodejů automobilů po měsících za určité období.
- Srovnání n-různých typů paliv n-různých automobilů po měsících za určité období.
- Srovnání bezpečnostních protokolů n-různých automobilů

Přehled aktuálně nabízených nových, či zánovních automobilů

Dalším aspektem, kterým se liší portály z kapitoly [1.1] a webové stránky zaměřené na přeprodej ojetých automobilů, je prodej a poptávka nových automobilů. Portály umožňují uživateli zvolit si konkrétní konfiguraci a následně vytvořit poptávku. Poptávka, v případě Gloffer, míří k ověřeným prodejcům. Na [Obrázek 5.7] je náhled stránky konfigurace z portálu Gloffer. Na této stránce se nachází odkaz na poptávku vozidla, která je umožněna až po registraci uživatele. Poptávka je zprostředkována pomocí kontaktního formuláře.



Obrázek 5.7: Pohled na stránku konfigurace Škoda CITIGO 5D z portálu Gloffer
Gloffer nenabízí výpis aktuálně dostupných nových, či zánovních automobilů. Na [Obrázek 5.8] je vidět stránka s výpisem aktuálně nabízených automobilů na portálu Carismo.

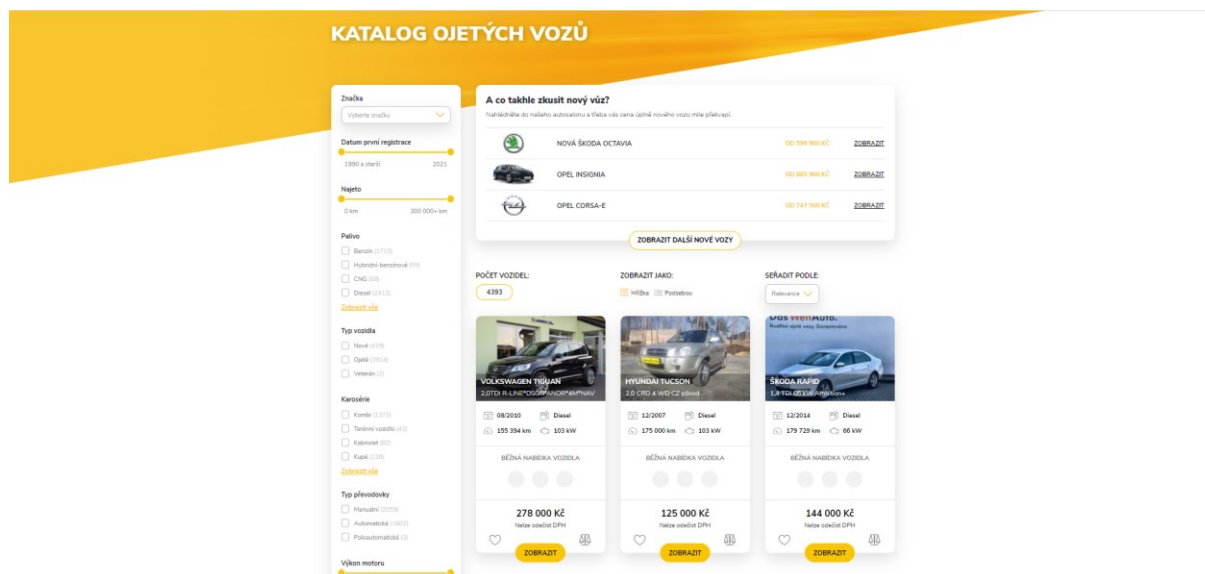


Obrázek 5.8: Pohled na stránku s výpisem aktuálně nabízených nových automobilů na portálu Carismo

Po důkladném zvážení vzhledem k analýze z kapitoly [2.], informací dostupných na portálech a možnosti dalších procesů uživatele se nebude seznam možných vizualizací statistik lišit od seznamu v podkapitole „Přehled konkrétního jednoho modelu, či jeho konfigurace“.

Přehled aktuálně nabízených ojetých automobilů

Portály z kapitoly [1.1] nenabízí uživatelům webu vkládat inzerát na ojetý automobil, jako tomu je v případě webových stránek zaměřených na přepravej ojetých automobilů, inzerce. Portály pouze agregují informace od partnerů. Kvalita a množství informací týkající se prodeje ojetého automobilu tak dodržují standard specifický pro jednotlivé portály. Na [Obrázek 5.9] je pohled na úvodní stránku s ojetými automobily portálu Gloffer.



Obrázek 5.9: Pohled na stránku s výpisem aktuálně nabízených ojetých automobilů na portálu Glosster

Vzhledem k analýze z kapitoly [2.1] lze pro ojeté vozy sestavit tento seznam dostupných vizualizací statistik.

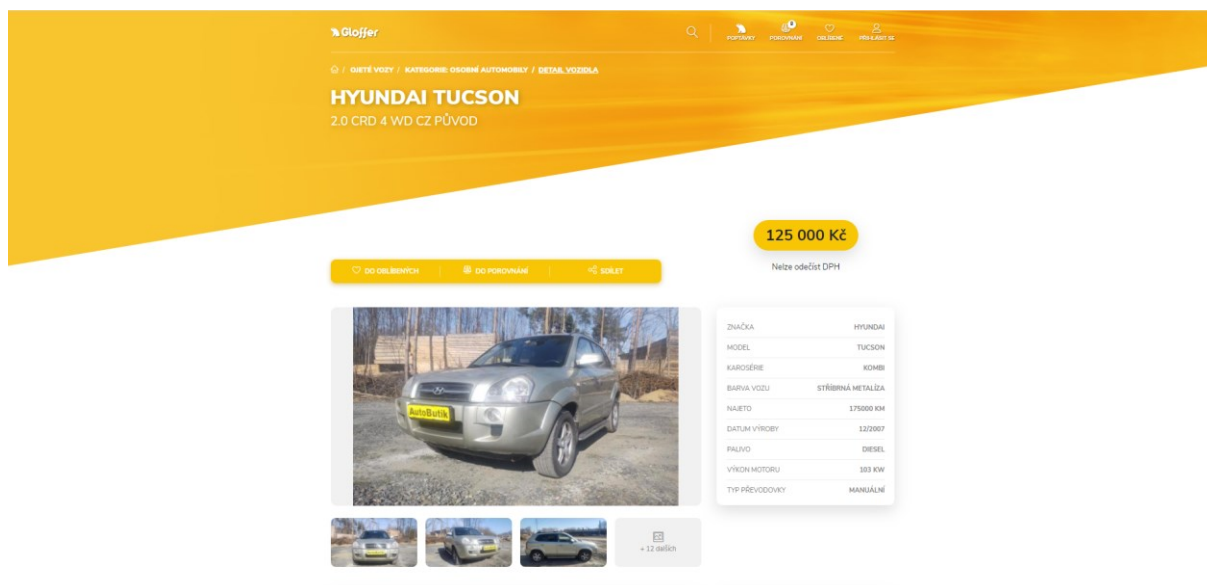
- Celkový prodej ojetých automobilů dle značky za určité období
- Prodej ojetých automobilů po měsících dle značky za určité období

Přehled nabídky ojetého vozidla

Pohled na stránku nabídky ojetého automobilu na [Obrázek 5.10]. Portály zmíněné v kapitole [1.1] obsahují velmi podobnou strukturu pro nabídku ojetého vozu. Typické prvky – blok s informacemi, cenou a fotografiemi jsou vizuálně dobře zpracovány s důrazem na přehlednost.

Z analýzy v kapitole [2.1], data týkající se ojetých vozů jsou značně omezená. Statistiky týkající se prodeje ojetých modelů automobilů nejsou dostupné. Statistiky zaznamenávají pouze prodej značek ojetých automobilů. Z tohoto důvodu zde může být použit seznam z podkapitoly „Přehled konkrétního jednoho modelu, či jeho konfigurace,“ jelikož se také jedná o konkrétní model, doplněný o tyto statistiky.

- Celkový prodej ojetých automobilů dle značky za určité období
- Prodej ojetých automobilů po měsících dle značky za určité období



Obrázek 5.10: Pohled na stránku s nabídkou ojetého automobilu

Vkládání dat z portálu SDA-CIA

Vkládání dat by měl být proces automatický, či poloautomatický s ohledem na mapování dat na již existující externí databázi. Vzhledem k analýze dat v [2.1], data jsou v tabulkovém souboru ve formátu .xls. Data je nutno buď přímo uživatelem nahrát do systému k jeho dalšímu zpracování, či spustit kód, který si potřebný soubor na portálu sda-cia nalezne a zpracuje.

Je žádoucí, aby byl uživatel informován o stavu jejich zpracování. Především je nutno uživatele informovat o nových entitách v tabulce (model, značka, skupina), případně o chybném zpracování částí tabulek.

Vkládání dat z portálu EuroNCAP

Z analýzy [2.2] zdroj dat EuroNCAP se nepodařilo nalézt algoritmus pro získání ID roku nutného ke zpracování dat prostřednictvím API. Proces je proto do jisté míry poloautomatický. Uživatel je nucen jednou ročně zjistit daný klíč a nastavit ho pro proces zpracování API.

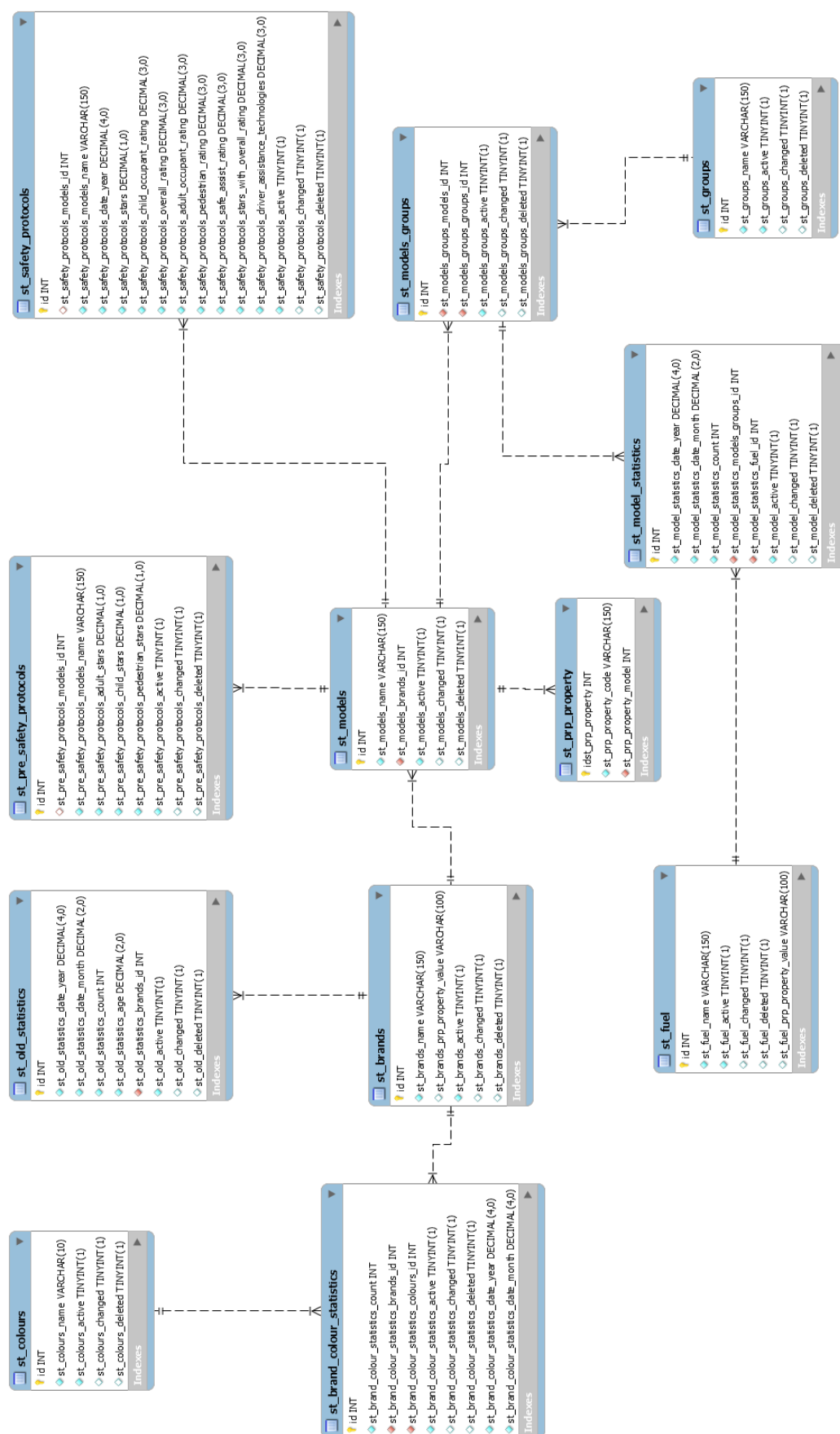
Data se na portálu EuroNCAP neobjevují pravidelně, je proto nutné nastavit si také vhodný interval mezi znovu vložení dat. Vzhledem k odlišnému dělení dat (modelů) je však nutné uživatele informovat o jeho případném novém vložení do databáze.

5.2 Datová analýza

V této podkapitole se nachází schéma relační databáze, které se využívá k uložení zvolených statistik a dotazování nad nimi. Je zde zápis entit a jejich vztahů.

Schéma relační databáze

Schéma lze vidět na [Obrázek 5.11].



Obrázek 5.11: Schéma relační databáze

Lineární zápis entit

Vysvětlivky: **Primární klíč**, cizí klíč, *unikátní klíč*

st_brands (**id**, st_brands_name, st_brands_prp_property_value, st_brands_active, st_brands_changed, st_brands_deleted)

st_brand_colour_statistics(**id**, st_brand_colour_statistics_count, st_brand_colour_statistics_brands_id, st_brand_colour_statistics_colours_id, st_brand_colour_statistics_active, st_brand_colour_statistics_changed, st_brand_colour_statistics_deleted, st_brand_colour_statistics_date_year, st_brand_colour_statistics_date_month)

st_colours (**id**, st_colours_name, st_colours_active, st_colours_changed, st_colours_deleted)

st_fuel (**id**, st_fuel_name, st_fuel_active, st_fuel_changed, st_fuel_deleted, st_fuel_prp_property_value)

st_groups (**id**, st_groups_name, st_groups_active, st_groups_changed, st_groups_deleted)

st_model_statistics (**id**, st_model_statistics_date_year, st_model_statistics_date_month, st_model_statistics_count, st_model_statistics_models_groups_id, st_model_statistics_fuel_id, st_model_active, st_model_changed, st_model_deleted)

st_models (**id**, st_models_name, st_models_brands_id, st_models_active, st_models_changed, st_models_deleted)

st_models_groups (**id**, st_models_groups_models_id, st_models_groups_groups_id, st_models_groups_active, st_models_groups_changed, st_models_groups_deleted)

st_old_statistics (**id**, st_old_statistics_date_year, st_old_statistics_date_month, st_old_statistics_count, st_old_statistics_age, st_old_statistics_brands_id, st_old_active, st_old_changed, st_old_deleted)

st_pre_safety_protocols (**id**, st_pre_safety_protocols_models_id, st_pre_safety_protocols_models_name, st_pre_safety_protocols_adult_stars, st_pre_safety_protocols_child_stars, st_pre_safety_protocols_pedestrian_stars, st_pre_safety_protocols_active, st_pre_safety_protocols_changed, st_pre_safety_protocols_deleted)

st_safety_protocols (**id**, st_safety_protocols_models_id, st_safety_protocols_models_name, st_safety_protocols_date_year, st_safety_protocols_stars, st_safety_protocols_child_occupant_rating, st_safety_protocols_overall_rating, st_safety_protocols_adult_occupant_rating, st_safety_protocols_pedestrian_rating, st_safety_protocols_safe_assist_rating, st_safety_protocols_stars_with_overall_rating, st_safety_protocols_driver_assistance_technologies, st_safety_protocols_active, st_safety_protocols_changed, st_safety_protocols_deleted)

st_prp_property (**id**st_prp_property, st_prp_property_code, st_prp_property_model)

Datový slovník

Tabulka: st_brands

Název	Datový typ (velikost)	NULL	Klíč
id	Int(11)	NO	PK
st_brands_active	tinyint(1)	NO	
st_brands_changed	Tinyint(1)	YES	
st_brands_deleted	Tinyint(1)	YES	
st_brands_name	varchar(150)	NO	
st_brands_prp_property_value	Varchar(150)	YES	

Tabulka B.1: Datový slovník st_brands

Tabulka: st_brand_colour_statistics

Název	Datový typ (velikost)	NULL	Klíč
id	Int(11)	NO	PK
st_brand_colour_statistics_active	tinyint(1)	NO	
st_brand_colour_statistics_brands_id	Int(11)	NO	FK
st_brand_colour_statistics_changed	tinyint(1)	YES	
st_brand_colour_statistics_colours_id	Int(11)	NO	FK
st_brand_colour_statistics_count	Int(11)	NO	
st_brand_colour_statistics_date_month	Decimal(2,0)	NO	
st_brand_colour_statistics_date_year	Decimal(4,0)	NO	
st_brand_colour_statistics_deleted	tinyint(1)	YES	

Tabulka B.2: Datový slovník st_brand_colour_statistics

Tabulka: st_colours

Název	Datový typ (velikost)	NULL	Klíč
id	Int(11)	NO	PK
st_colours_active	tinyint(1)	NO	
st_colours_changed	Tinyint(1)	YES	
st_colours_deleted	Tinyint(1)	YES	
st_colours_name	Varchar(10)	NO	

Tabulka B.3: Datový slovník st_colours

Tabulka: st_fuel

Název	Datový typ (velikost)	NULL	Klíč
id	Int(11)	NO	PK
st_fuel_active	tinyint(1)	NO	
st_fuel_changed	Tinyint(1)	YES	
st_fuel_deleted	Tinyint(1)	YES	
st_fuel_name	varchar(150)	NO	
st_fuel_prp_property_value	Varchar(100)	YES	

Tabulka B.4: Datový slovník st_fuel

Tabulka: st_groups

Název	Datový typ (velikost)	NULL	Klíč
id	Int(11)	NO	PK
st_groups_active	Tinyint(1)	NO	
st_groups_changed	Tinyint(1)	YES	
st_groups_deleted	Tinyint(1)	YES	
st_groups_name	Varchar(150)	NO	

Tabulka B.5: Datový slovník st_groups

Tabulka: st_model_statistics

Název	Datový typ (velikost)	NULL	Klíč
id	int(11)	NO	PK
st_model_active	tinyint(1)	NO	
st_model_changed	tinyint(1)	YES	
st_model_deleted	tinyint(1)	YES	
st_model_statistics_count	int(11)	NO	
st_model_statistics_date_month	Decimal(2,0)	NO	
st_model_statistics_date_year	Decimal(4,0)	NO	
st_model_statistics_fuel_id	int(11)	NO	FK
st_model_statistics_models_groups_id	int(11)	NO	FK

Tabulka B.6: Datový slovník st_model_statistics

Tabulka: st_models

Název	Datový typ (velikost)	NULL	Klíč
id	int(11)	NO	PK
st_models_active	tinyint(1)	NO	
st_models_brands_id	Int(11)	NO	FK
st_models_changed	tinyint(1)	YES	
st_models_deleted	tinyint(1)	YES	
st_models_name	Varchar(150)	NO	

Tabulka B.7: Datový slovník st_models

Tabulka: st_models_groups

Název	Datový typ (velikost)	NULL	Klíč
id	int(11)	NO	PK
st_models_groups_active	tinyint(1)	NO	
st_models_groups_changed	Tinyint(1)	YES	
st_models_groups_deleted	tinyint(1)	YES	
st_models_groups_groups_id	int(11)	NO	FK
st_models_groups_models_id	int(11)	NO	FK

Tabulka B.8: Datový slovník st_models_groups

Tabulka: st_old_statistics

Název	Datový typ (velikost)	NULL	Klíč
id	int(11)	NO	PK
st_old_active	tinyint(1)	NO	
st_old_changed	tinyint(1)	YES	
st_old_deleted	tinyint(1)	YES	
st_old_statistics_age	Decimal(2,0)	NO	
st_old_statistics_brands_id	int(11)	NO	FK
st_old_statistics_count	int(11)	NO	
st_old_statistics_date_month	Decimal(2,0)	NO	
st_old_statistics_date_year	Decimal(4,0)	NO	

Tabulka B.9: Datový slovník st_old_statistics

Tabulka: st_pre_safety_protocols

Název	Datový typ (velikost)	NULL	Klíč
id	int(11)	NO	PK
st_pre_safety_protocols_active	tinyint(1)	NO	
st_pre_safety_protocols_adult_stars	Decimal(1,0)	NO	
st_pre_safety_protocols_changed	tinyint(1)	YES	
st_pre_safety_protocols_child_stars	Decimal(1,0)	NO	
st_pre_safety_protocols_deleted	tinyint(1)	YES	
st_pre_safety_protocols_models_id	int(11)	YES	FK
st_pre_safety_protocols_models_name	Varchar(150)	NO	
st_pre_safety_protocols_pedestrian_stars	Decimal(1,0)	NO	

Tabulka B.10: Datový slovník st_pre_safety_protocols

Tabulka: st_prp_property

Název	Datový typ (velikost)	NULL	Klíč
idst_prp_property	int(11)	NO	PK
st_prp_property_code	varchar(150)	NO	

Tabulka B.11: Datový slovník st_prp_property

Tabulka: st_safety_protocols

Název	Datový typ (velikost)	NULL	Klíč
id	int(11)	NO	PK
st_safety_protocols_active	tinyint(1)	NO	
st_safety_protocols_adult_occupant_rating	Decimal(3,0)	NO	
st_safety_protocols_changed	tinyint(1)	YES	
st_safety_protocols_child_occupant_rating	Decimal(3,0)	NO	
st_safety_protocols_date_year	Decimal(4,0)	NO	
st_safety_protocols_deleted	tinyint(1)	YES	
st_safety_protocols_driver_assistance_technologies	Decimal(3,0)	NO	
st_safety_protocols_models_id	int(11)	YES	FK
st_safety_protocols_models_name	Varchar(150)	NO	
st_safety_protocols_overall_rating	Decimal(3,0)	NO	
st_safety_protocols_pedestrian_rating	Decimal(3,0)	NO	
st_safety_protocols_safe_assist_rating	Decimal(3,0)	NO	
st_safety_protocols_stars	Decimal(1,0)	NO	
st_safety_protocols_stars_with_overall_rating	Decimal(3,0)	NO	

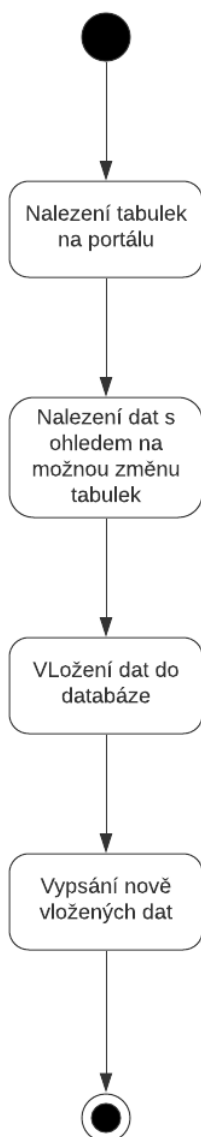
Tabulka B.12: Datový slovník st_safety_protocols

5.3 Stavová analýza importu a dotazování se dat pro účel vizualizace

V této podkapitole bude vysvětlen zjednodušený postup algoritmu importu ze zdrojů dat v kapitole [2]. Bude zde představen návrh algoritmu pro práci s daty a jejich vizualizaci s ohledem na jednoduchou znovupoužitelnost.

Stavová analýza importu z portálu SDA-CIA

Z analýzy zdrojů dat lze navrhnout plánování importu v intervalu jednoho měsíce. Proces to tak nebude často opakovaný – jednou měsíčně a bude lehce plánovatelný – poslední den v měsíci v méně vytíženou dobu webového serveru. Proces importu tabulek se dá zobecnit do čtyř kroků při plné automatizaci (bez nutnosti uživatelem nahrát soubor), zjednodušený diagram stavů lze vidět na [Obrázek 5.12].



Obrázek 5.12: Zjednodušený *diagram stavů*

Uživatel nebo program spustí program pro import.

Prvním krokem je nalezení tabulek na adrese „<https://www.sda-cia.cz/repository-volnedostupna>“, kdy ke hledání je využito nalezení odkazů – HTML tagů <a> s textem sestaveným dle jasně dané struktury vyplývající z analýzy [2.1] s pomocí PHP Simple HTML Parser [5.1].

V druhém kroku, je třeba načíst tabulky ze souboru pomocí rozšíření PHPSpreadsheet. Nalézt data s ohledem na nastavené hranice z externího souboru ve formátu JSON. Zpracovat samotná data do přehledného serializovatelného objektu.

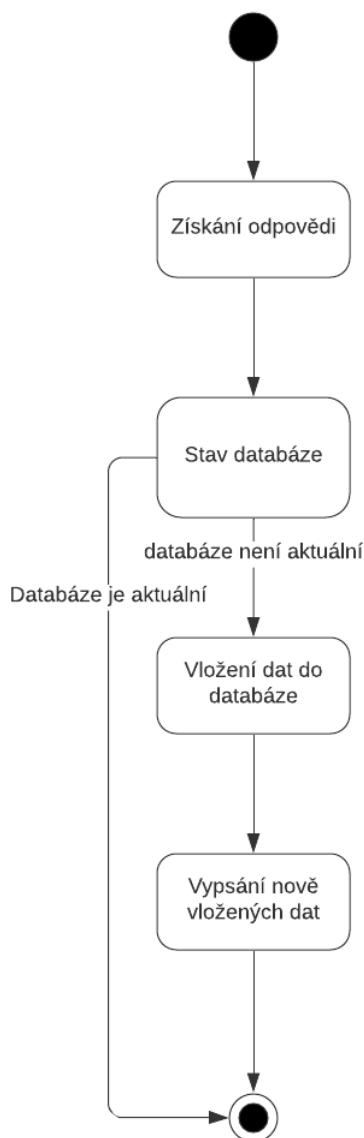
Třetí krok je serializace objektů a zaslání do databáze. Ke správnému vložení do databáze a ušetření množství přenesených dat je prostřednictvím funkce v databázi serializovaný objekt zpracován. Funkce databáze vyhodnotí, zda vložit objekt, či jej už uložený má. Jako odpověď vrátí nově vložené objekty ve formátu JSON.

Ve čtvrtém kroku program zpracuje odpověď z databáze a vypíše nově vložené objekty.

Program se ukončí.

Stavová analýza importu dat EuroNCAP

Z analýzy v kapitole [2.2] není patrný interval přírůstku nových bezpečnostních protokolů. Proces to tak může být těžko plánovatelný. Počet nových protokolů je však vždy v řádů jednotek. Lze si tak stanovit interval – jeden den, či jako v případě tabulek z SDA-CIA – jednou měsíčně. Celý proces tak lze opět zjednodušit do tří kroků. Zjednodušený diagram stavů lze vidět na [Obrázek 5.13].



Obrázek 5.13: Zjednodušený *diagram stavů*

Uživatel nebo program spustí program pro import.

V prvním kroku je třeba získat odpověď z API portálu EuroNCAP dostupné na adrese „<https://www.euroncap.com/Umbraco/EuroNCAP/SearchApi/GetAssessmentSearch>“ s příslušnými parametry.

Ve druhém kroku je třeba zjistit, zda je databáze aktuální. Dojde k porovnání počtu výsledků – pokud je v odpovědi stejný počet záznamů jako v databázi, stav databáze je aktuální a program se může ukončit. Pokud je počet záznamů v databázi menší, než v odpovědi pokračuj ke třetímu kroku.

Třetí krok je opět vložení do databáze, kdy je použit stejný princip jako v předchozí kapitole – serializuj objekt a pokus se jej vložit do databáze pomocí jasně dané funkce v databázi. Funkce se postará o rozhodnutí, zda záznam vložit, či již v tabulce existuje.

Ve čtvrtém kroku program zpracuje odpověď databáze – JSON a vypíše, zda-li je záznam nový, či již v databázi existoval.

Program se ukončí.

Dotazování se dat pro účel vizualizace

Vzhledem k možným využití komponenty a s ohledem na její možný rozvoj vzniknou dvě třídy. Jedna třída se bude starat o data a práce s nimi, druhá třída se bude starat o vykreslení komponenty.

První třída bude mít také za úkol:

- Zpracovat parametry pro zobrazení (ověření a parsování).
- Dotázat se o data databázi.
- Přeprocovat data do srozumitelného, setříděného klíč – hodnota pole.

Ve třídě dojde pomocí metody k nastavení parametrů, dle doporučeného postupu – URL adresy sestavené pomocí GET požadavku. Část adresy tzv. Query je přepracovaná na JSON objekt, který je následně ošetřen proti možným nežádoucím vstupům uživatele.

Po získání JSON objektu s parametry se program dotazuje databáze obsahující proceduru, do které stačí vložit JSON objekt. Procedura se sama rozhodne, jaký SQL dotaz má vykonat a tím také jaká data vrátit. Program získá odpověď na dotaz databáze. V případě dotazů na bezpečnost není potřeba se dotazovat pomocí procedury, jelikož se jedná o jednoduchý SQL dotaz bez větší logiky.

Program se rozhodne dle vstupních parametrů, jak data seskupit do vhodné struktury klíč – hodnota.

Druhá třída starající se o vykreslení grafů bude mít za úkol:

- Vložit potřebné závislosti pro vykreslení.
- Vykreslit potřebné HTML tagy spolu s jejich vlastnostmi – parametry a přidružené CSS z externího souboru.
- Sestavit a vykreslit správně přidružený Javascript pro potřeby grafu.
- Sestavit a vykreslit tabulku.

Komponenta bude obsahovat metodu pro vrácení všech potřebných závislostí ke korektnímu fungování webové stránky. To zahrnuje knihovnu jQuery, závislosti frameworku Bootstrap 4 a ho samotného, knihovnu Chartjs spolu s pluginem Datalabels.

Sestavení grafů se skládá ze dvou částí. První je vykreslení HTML tagů, konkrétně tagu <canvas> a jeho obalení <div> tagem pro manipulaci s ním. Třída musí podporovat úpravu tagů ve smyslu přidání třídy, nastavení stylu elementu.

Při sestavování Javascriptu pro účel vykreslení grafu komponenta správně sestaví potřebný kód s ohledem na data. Při špatném zadání informuje uživatele o špatných parametrech, či případné neexistenci dat.

Pro účel sestavení tabulky si komponenta sama zajistí přeskupení dat pro potřeby tabulky a následně vykreslí spolu se styly z externího souboru. Sestavené záznamy v tabulce jsou zároveň odkazy pro sestavení grafů.

5.4 Funkční analýza

V této podkapitole je funkční analýza importů dat ze zdrojů. Tato analýza slouží k určení seznamu potřebných funkcí databáze. Jsou zde popsány funkce dvou nejdůležitějších procesů – vkládání dat do databáze z SDA-CIA a EuroNCAP. Vzhledem k rozsahu zdrojového kódu většiny funkcí zde nebude vypsán kód, ale pouze popis, vstupy a výstupy.

Seznam funkcí pro import dat z SDA-CIA

Zodpovědnost: Správce / Systém

- insertColoursStatisticJson – Procedúra pro vložení statistických dat barev značek
- InsertModelGroupStatisticJson – Procedúra pro vložení statistických dat modelů
- insertOldStatisticJson – Procedúra pro vložení statistických dat ojetin
- insertBrandsJson – Funkce pro vložení značek
- insertColoursJson – Funkce pro vložení barev
- insertFuelTypesJson – Funkce pro vložení paliv
- insertGroupJson – Funkce pro vložení tříd
- insertModelGroupJson – Funkce pro vložení model – třída

Seznam funkcí pro import dat z EuroNCAP

Zodpovědnost: Správce / Systém

- insertSafetyProtocol – Procedúra pro vložení bezpečnostních protokolů
- insertPreSafetyProtocol – Procedúra pro vložení starších bezpečnostních protokolů

insertColoursStatisticJson

Tato procedúra má na starost vkládání statistických dat barev do databáze. Procedúra sama nalezne, potřebná ID k vytvoření záznamu a záznam vytvoří.

Vstup je Serializovaný JSON obsahující data v následujícím pořadí, kdy n je číslo záznamu pro vložení v pořadí od 0:

1. Rok – DECIMAL(4,0)
2. Měsíc – DECIMAL(2,0)
- 3+n. Značka záznamu – VARCHAR(150)
- 4+n. Barva záznamu – VARCHAR(150)
- 5+n. Počet prodaných kusů - INT

InsertModelGroupStatisticJson

Tato procedúra má na starost vkládání statistických dat modelů do databáze. Procedúra sama nalezne, potřebná ID k vytvoření záznamu a záznam vytvoří.

Vstup je Serializovaný JSON obsahující data v následujícím pořadí, kdy n je číslo záznamu pro vložení v pořadí od 0:

1. Rok – DECIMAL(4,0)
2. Měsíc – DECIMAL (2,0)
3. Třída pro všechny záznamy – VARCHAR(150)
- 4+n. Jméno modelu záznamu – VARCHAR(150)
- 5+n. Jméno paliva záznamu – VARCHAR(150)
- 6+n. Počet prodaných kusů - INT

insertOldStatisticJson

Tato procedúra má na starost vkládání statistických dat ojetin do databáze. Procedúra sama nalezne, potřebná ID k vytvoření záznamu a záznam vytvoří.

Vstup je Serializovaný JSON obsahující data v následujícím pořadí, kdy n je číslo záznamu pro vložení v pořadí od 0:

1. Rok – DECIMAL(4,0)
2. Měsíc – DECIMAL(2,0)
- 3+n. Značka záznamu – VARCHAR(150)
- 4+n. Stáří záznamu – DECIMAL(2,0)
- 5+n. Počet prodaných kusů - INT

insertBrandsJson

Tato funkce má na starost vkládání značek do databáze. Funkce se sama rozhodne, zda-li značka již v databázi existuje, či je třeba vytvořit nový záznam. Funkce vrátí JSON s nově vytvořenými záznamy.

Vstup je Serializovaný JSON obsahující data v následujícím pořadí, kdy n je číslo záznamu pro vložení v pořadí od 0:

- n. Název značky - VARCHAR(150)

Výstup:

JSON s názvy značek.

insertColoursJson

Tato funkce má na starost vkládání barev do databáze. Funkce se sama rozhodne, zda-li barva již v databázi existuje, či je třeba vytvořit nový záznam. Funkce vrátí JSON s nově vytvořenými záznamy.

Vstup je Serializovaný JSON obsahující data v následujícím pořadí, kdy n je číslo záznamu pro vložení v pořadí od 0:

- n. Název barvy – VARCHAR(150)

Výstup:

JSON s názvy barev.

insertFuelTypesJson

Tato funkce má na starost vkládání typů paliv do databáze. Funkce se sama rozhodne, zda-li typ paliva již v databázi existuje, či je třeba vytvořit nový záznam. Funkce vrátí JSON s nově vytvořenými záznamy.

Vstup je Serializovaný JSON obsahující data v následujícím pořadí, kdy n je číslo záznamu pro vložení v pořadí od 0:

- n. Název typu paliva – VARCHAR(150)

Výstup:

JSON s názvy typů paliv.

insertGroupsJson

Tato funkce má na starost vkládání tříd do databáze. Funkce se sama rozhodne, zda-li třída již v databázi existuje, či je třeba vytvořit nový záznam. Funkce vrátí JSON s nově vytvořenými záznamy.

Vstup je Serializovaný JSON obsahující data v následujícím pořadí, kdy n je číslo záznamu pro vložení v pořadí od 0:

n. Název třídy – VARCHAR(150)

Výstup:

JSON s názvy tříd.

insertModelGroupJson

Tato funkce má na starost vkládání modelů do databáze. Funkce se sama rozhodne, zda-li model již v databázi existuje, či je třeba vytvořit nový záznam. Funkce zároveň zajišťuje konzistenci databáze díky sjednocení názvů značek, modelů – pouze malá písmena bez mezer. Funkce vrátí JSON s nově vytvořenými záznamy.

Vstup je Serializovaný JSON obsahující data v následujícím pořadí, kdy n je číslo záznamu pro vložení v pořadí od 0:

1. Název třídy pro všechny záznamy v JSONu – VARCHAR(150)

2+n. Název modelu pro záznam – VARCHAR(150)

3+n. Název značky záznamu – VARCHAR(150)

Výstup:

JSON s názvy modelů.

insertSafetyProtocol

Tato procedura má na starost vkládání statistických dat starších bezpečnostních protokolů do databáze. Procedura sama rozhodne, zda-li má záznam vložit a pokusí se jej namapovat.

Vstup jsou parametry v pořadí:

1. Název modelu – VARCHAR(150)

2. Rok – DECIMAL(4,0)

3. Počet hvězdiček dle EuroNCAP – DECIMAL(1,0)

4. Hodnocení bezpečnosti dětí – DECIMAL(3,0)

5. Hodnocení bezpečnosti celkově – DECIMAL(3,0)

6. Hodnocení bezpečnosti dospělí – DECIMAL(3,0)

7. Hodnocení bezpečnosti řidič – DECIMAL(3,0)

8. Hodnocení asistentů – DECIMAL(3,0)

9. Hodnocení celkově s hvězdičkama – DECIMAL(3,0)

10. Hodnocení technologií asistentů – DECIMAL(3,0)

insertPreSafetyProtocol

Tato procedura má na starost vkládání statistických dat starších bezpečnostních protokolů do databáze. Procedura sama rozhodne, zda-li má záznam vložit a pokusí se jej namapovat.

Vstup jsou parametry v pořadí:

1. Název modelu – VARCHAR(150)

2. Hodnocení bezpečnosti dětí – DECIMAL(1,0)

3. Hodnocení bezpečnosti dospělí – DECIMAL(1,0)

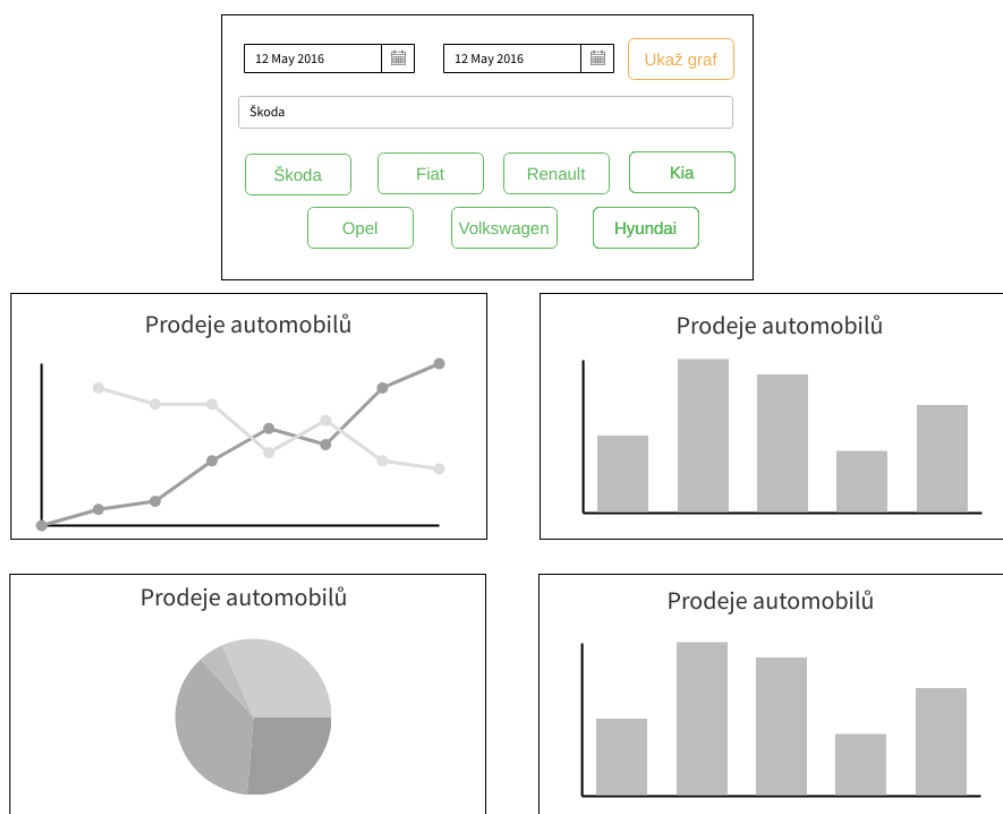
4. Hodnocení bezpečnosti řidič – DECIMAL(1,0)

5.5 Návrh drátěného modelu

V této podkapitole nalezneme návrhy drátěného modelu pro vybrané uživatelské rozhraní komponenty. Uživatelské rozhraní pro náhled modelu, značky, třídy, náhled rozhraní pro uživatelem nastavitelnou vizualizaci dat.

Návrh rozhraní pro náhled modelu, značky a třídy

Drátěný model pro všechny velké náhledy (model, značka, třída) je navržen tak, aby byl co nejprehlednější. Dělí se do pěti boxů, kdy v prvním boxu lze nastavit parametry pro zobrazování grafů spolu s odkazy na další relevantní náhledy. Každý náhled má stejnou strukturu pěti boxů, první box se liší pouze v možnosti zvolení konkrétního náhledu (značky pro náhled značky, model pro náhled modelu, třídy pro náhled třídy) a relevantních odkazů. Další odlišnost náhledu jsou typy grafů a data v boxech pro grafy. Z tohoto důvodu je zde uveden pouze jeden drátěný model pro všechny tři velké náhledy. Drátěný model lze vidět na [obrázek 5.14].

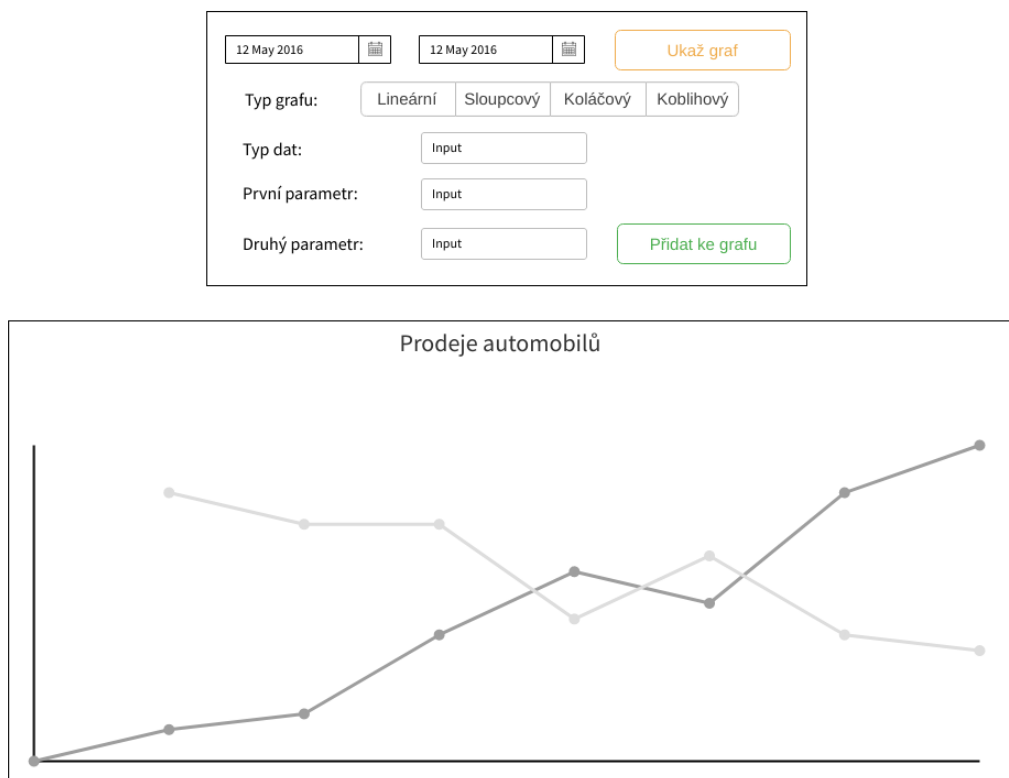


Obrázek 5.14: Drátěný model pro náhled modelu

Náhled rozhraní pro uživatelem nastavitelnou vizualizaci dat

Kvůli velkému množství dat, která lze na základě zpracovaných statistik vizualizovat, vznikla potřeba uživatelsky intuitivního, nastavitelného náhledu. Uživatel tak není vázán na implementované

grafy s omezenou možností zobrazení dat, ale může si sám zvolit typ grafu, pohled nad daty, konkrétní data a případně také přidávat dimenze do grafu. Drátěný model lze vidět na [Obrázek 5.15].



Obrázek 5.15: Drátěný model pro nastavitelnou manuální vizualizaci dat

Návrh drátěného modelu pro rozcestník

Správce komponenty by měl mít možnost jednoduše spustit kód souboru prostřednictvím odkazu. Zároveň je vhodné, aby byl správce informován, zda jsou data aktuální, či je třeba nahrát data ze zdrojů. Pro tento účel je navržen drátěný model na [Obrázek 5.16]. Stránka pomocí odkazů nabízí správci rychlý přístup k implementovaným náhledům a velkým manuálním statistik.

Import dat	Rozcestník náhledů
Automatický import dat z portálu SDA-CIA ZDE	Náhled modelu
Automatický import dat z portálu EuroNCAP ZDE	Náhled značky
Stav dat v databázi - zdroj dat	Náhled třídy
Data z portálu SDA-CIA pro tento měsíc vloženy: ANO	Velké manuální statistiky
Data z EuroNCAP jsou aktuální: ANO	

Obrázek 5.16: Drátěný model pro rozcestník

6 Návrh implementace

V této kapitole jsou uvedeny použité technologie. Každá technologie je zde představena a v případě databáze a programovacího jazyka na webovém serveru také odůvodněna. Kapitola obsahuje schéma relační databáze, lineární zápis jejich entit a vztahů. Jsou představeny návrhy drátěného modelu uživatelského rozhraní a návrh postupu importu a dotazování se dat pro účel vizualizace. Poslední drátěný model je návrh zobrazení stránky pro účel prezentace výsledné práce, slouží jako rozcestník pro správce, aby mohl celou komponentu sám otestovat, importovat data a vizualizovat grafy nebo náhledy.

6.1 Návrh technologií

V této podkapitole jsou představeny použité technologie a případně také jejich vybraná rozšíření.

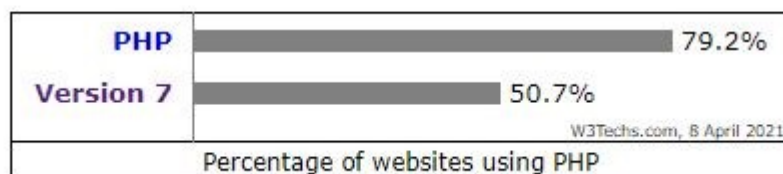
PHP

PHP je univerzální skriptovací jazyk, vhodný především pro vývoj webových aplikací. Informace o webovém serveru zde [21]. Celým názvem Hypertext Preprocessor je programovací jazyk, který dle webu W3Techs využívá 79.2% všech webových serverů. [8] Z [Obrázek 6.1] je také patrné většinového zastoupení PHP 7. Webové servery, většinou Apache, jsou tak velmi dobře dostupné. S tak velkým poměrovým zastoupením je patrná velká komunita programátorů zabývajících se PHP. Díky tomu existuje také mnoho dostupných knihoven zdarma. K instalaci a správě knihoven se často využívá správce závislostí, Composer nebo Packagist v případě PHP. Ten programátorovi usnadňuje práci a konzistentní verzování knihoven. Tito správci umožňují jednoduchou správu závislostí pomocí terminálu. Mezi znalosti programátora implementujícího závislosti, tak jsou třeba také příkazy ke správě těchto nástrojů.

How to read the diagram:

PHP is used by 79.2% of all the websites.

Version 7 is used by 50.7% of all the websites.



Obrázek 6.1: Procentální zastoupení programovacího jazyka ve webových aplikacích dle W3Techs

Tento jazyk je tak vhodný nejen pro běh samostatných webových aplikací, ale také například ke nalezení obsahu na webové stránce, stažení souborů, či zpracování tabulek.

Pro zpracování tabulek je použita knihovna PhpSpreadsheet, knihovna napsaná pouze v PHP nabízející řadu tříd pro čtení a zápis rozličných souborových formátů jako je Excel a LibreOffice Calc. [9]. Díky knihovně je tak snadné zpracovat tabulky různých formátů konzistentním způsobem. Knihovna vyžaduje PHP 7.2 nebo vyšší. Další potřebné PHP závislosti jsou obsaženy v souboru composer.json, o který se stará správce závislostí Composer.

K usnadnění práce s webovou stránkou je použit volně dostupný skript PHP Simple HTML DOM Parser. Tento skript nevyžaduje žádné další závislosti. Byl napsán pro PHP verzi 5, jeho použití je však ještě dnes aktuální. Podporuje invalidní HTML kód. Díky tomuto parseru je nalezení požadovaného HTML tagu a jeho vlastností otázkou dvou řádků kódu. [10]

MySQL

MySQL je open-source transakční systém pro správu relačních databází. Systém využívají největší společnosti zabývající se činností spojenou s webovými aplikacemi. Mezi ně patří např. Facebook, Youtube, Paypal, Twitter a mnoho dalších. MySQL vytvořila švédská společnost MySQL AB za účelem efektivní a spolehlivé možnosti pro správu dat domácím i profesionálním uživatelům.

V systému jsou data strukturovaná jako tabulky, konkrétní záznam pak reprezentují řádky tabulky. Pro dotazování dat se využívá SQL, či PL/SQL pro procedurální práci s tabulkami. Více o MySQL zde [22]

HTML

Hypertext Markup Language je značkovací jazyk, jádrem každé webové aplikace. Značky – tagy, označují části stránky, například záhlaví, obrázek, či odkaz na jinou stránku. Pomocí značkovacího jazyka lze jasně definovat vzhled a strukturu dokumentu – webové stránky, PDF, mobilních aplikací atp. Více o tom, co to je spolu se seznamem vybraných HTML tagů zde [23]. Aktuální verze HTML je verze HTML5, která již podporuje více, než jen strukturu a vzhled dokumentu, např. otevření klienta pro odeslání emailu.

CSS

Cascading Style Sheets je jazyk, který se používá k popsání prezentace dokumentu napsaného v HTML nebo XML. CSS popisuje, jak by měly jednotlivé elementy být vykresleny na obrazovku, papír, či jiná média.[11]

CSS je součástí základních jazyků otevřeného webu a je standardem pro všechny prohlížeče. Oficiálně aktuální verze je CSS3, ačkoliv již existuje i novější verze. Místo vydávání nových verzí CSS a změnou práce s nimi se vydávají pouze nová doporučení, jak s CSS a jeho moduly pracovat.

Při práci s CSS a responzivním designem je vhodné užití frameworku, který usnadní zobrazování na různých zařízeních. Bootstrap je framework usnadňující práci při implementaci responzivního vzhledu. Obsahuje jak předdefinované CSS, tak JS balíček s předdefinovým interaktivním chováním některých prvků. K použití balíčku s JS je nutno také přiložit knihovnu jQuery. Pro práci s responzivním designem využívá mřížkový systém. Aktuální beta verze je verze 5, je však vhodné použití více otestované verze, verze 4.

JavaScript

JavaScript, zkráceně JS, je multiplatformní, objektově orientovaný, událostmi řízený skriptovací jazyk podporovaný v moderních prohlížečích.[12] Díky kterému lze naprogramovat libovolné chování, ať už pro interakci s uživatelem, či nejrůznějších animací, validací, asynchronních požadavků apod.

Pro usnadnění implementace některých typických úkonů se využívá rozšíření jQuery, který implementuje např. snadnou iteraci přes objekt, práci s poli atp. Jak již bylo zmíněno v předchozí kapitole, k fungování JS balíčku ve frameworku Bootstrap 4 je zapotřebí přiložit i tuto knihovnu. Není proto důvod, proč knihovnu nevyužít také mimo framework Bootstrap 4.

Při práci s grafy je použita knihovna Chart.js, která byla zvolena na základě představení v kapitole [1.2]. Knihovna obsahuje velké množství již připravených animací a interakcí událostmi myši. Celou práci tak výrazně usnadňuje. Na požadavek vizuálně přitažlivějšího zobrazování nápověd při interakci s myší je použit plugin Datalabels rozšiřující knihovnu o chování a vzhled nápověd grafu.

6.2 Shrnutí návrhu implementace

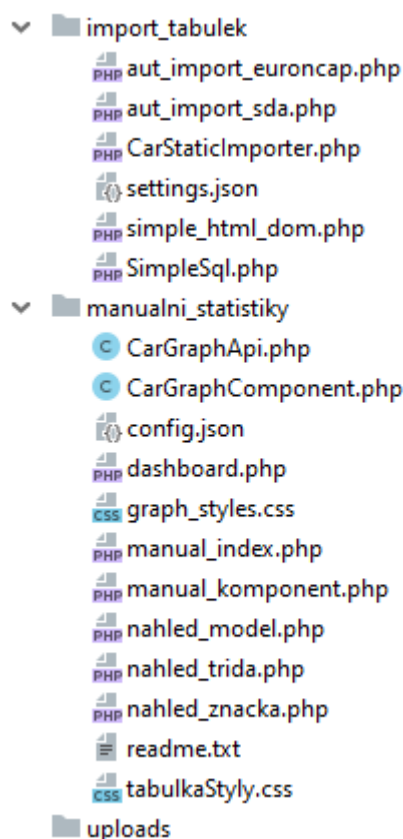
Celé řešení bude sestaveno na technologiích představených v první podkapitole běžících na operačním systému Windows. Webový server Apache2.4 s nastavenými rozšířeními pro práci s Mysqli, cURL a komunikaci pomocí SSL. Více informací o webovém serveru Apache zde [24]. Řešení bude implementováno na PHP verzi 7.4.11 NTS. Je nutno zvětšit parametr `max_execution_time` a `upload_max_filesize` v souboru `PHP.ini`. Pro délku spuštění skriptu postačuje hodnota větší než 300, velikost nahraných tabulek dosahuje maximální velikosti 2,7MB, proto je nutno navýšit také `upload_max_filesize` na hodnotu alespoň „4M“. Databázový server MySQL Community Server v8.0.23 plně vyhovuje potřebám aplikace. S použitím HTML5 a stylů CSS3 budou generovány základní elementy pro vykreslení grafů a samotné tabulky. Pro dosažení interaktivního chování, responzivního vzhledu bude použit framework Bootstrap 4.4.1 s JavaScript knihovnou jQuery 3.5.1. K vykreslování grafů je použita Javascipt knihovna Chart.js 2.8.0 s pluginem Datalabels v1.0.0.

7 Implementace

Program je implementován v prostředí PHPStorm pomocí jazyka PHP a v prostředí MySQL Workbench 8 pro implementaci databáze pomocí SQL a PL/SQL. V kapitole je uvedena struktura výsledného programu, výběr fragmentu PHP kódu pro použití komponenty, fragment procedury pro výběr dat zobrazení grafu a seznámení s implementovanými možnostmi procedury.

7.1 Struktura Programu

Celý program složený z .php souborů, souborů s příponou .json a souborů s příponou .css je rozdělen do dvou složek. Třetí složka uploads slouží pro nahrání tabulek z portálu SDA-CIA. Strukturu lze vidět na [Obrázek 7.1]. Ve složce import_tabulek se nachází všechny soubory týkající se importu dat. Složka manualni_statistiky představuje všechny soubory pro komunikaci s databází pro účely zobrazení dat, soubory se třídou a rozložením náhledů pro vykreslení grafů, konfiguračních souborů a souborů obsahující kaskádové styly pro potřeby náhledů a tabulek.



Obrázek 7.1: Struktura *adresáře*

Seznam s krátkým popisem souborů ve složce import_tabulek:

- aut_import_euroncap.php - Automatický import bezpečnostních protokolů
- aut_import_sda.php – Automatický import tabulek a stažení tabulek z portálu SDA-CIA
- CarStaticImporter.php – Soubor s třídou CarStaticImporter pro import .xls tabulek.
- SimpleSql.php – Soubor s interfacem IData a třídou SimpleSql implementující požadovaný interface pro práci s třídou CarStaticImporter.
- simple_html_dom.php – Soubor s funkcema pro usnadnění práce s HTML dokumentem.

- setting.json – Soubor obsahující nastavení pro potřeby importu dat ve formátu JSON.
- Seznam s krátkým popisem souborů ve složce import_tabulek:
- CarGraphApi.php – Soubor s třídou CarGraphApi pro komunikaci s databází, validací parametrů a transformací dat.
- CarGraphComponent.php – Soubor s třídou CarGraphComponent pro vykreslování grafů, tabulek a vkládání závislostí v HTML dokumentu.
- manual_index.php – Soubor pro vykreslení plně nastavitelných statistik.
- manual_komponent.php – Soubor pro vykreslení samostatného grafu dle zadané parametrů.
- nahled_model.php – Soubor pro vykreslení velkého náhledu pro model.
- nahled_znacka.php – Soubor pro vykreslení velkého náhledu pro značku.
- nahled_trida.php – Soubor pro vykreslení velkého náhledu pro třídu.
- dashboard.php – Soubor pro vykreslení rozcestníku programu.
- graph_styles.css – Soubor obsahující kaskádové styly určené k použití s grafy.
- tabulkaStyly.css - Soubor obsahující kaskádové styly určené k použití pro tabulky.
- readme.txt – Soubor obsahující poznámky programátora pro zobrazování komponenty.

7.2 Důležité algoritmy v podobě zdrojových kódů

Ukázka použití PHP komponenty

Všechn kód je volně dostupný. Pro účel této podkapitoly byl vybrán PHP kód s použitím komponenty – získání dat z databáze a vykreslení jednoho grafu a jeho tabulky. Nebude zde ukázaná implementace samotné komponenty, nýbrž její implementace pro případné použití na webové stránce.

Implementace ukázky použití komponenty by měla být jednoduchá a zároveň ukázat čtenáři vše potřebné pro správné fungování vykreslení jednoho zvoleného grafu. Pro ukázku si zvolíme sloupcový graf s jednou dimenzí, graf bude zobrazovat všechna dostupná paliva pro značku Audi za období od ledna 2020 do listopadu 2020.

```
$CarGraphApi = new CarGraphApi();
$url = "http://www.testovani.cz?graphType=bar&odDate=2020-01&doDate=2020-12&object[]=Audi&type[]=new_brands_of_brand_fuels&";
$CarGraphApi->setParametersByUrl($url);
$GraphComponent = new CarGraphComponent($CarGraphApi);
```

V momentě, kdy máme implementován kód z ukázky výše je připraveno vše potřebné pro volání vykreslení řetězce s závislostmi, HTML tagu <canvas> pro potřeby grafu, JavaScript kódu potřebný k vykreslení grafu a vykreslení tabulky.

```
echo $GraphComponent->getHeadLinks();
echo $GraphComponent->getCanvas("example_graph");
echo $GraphComponent->getGraph("example_graph");
echo $GraphComponent->getTable();
```


Vypsání těchto řetězců je třeba správně umístit do struktury stránky, např. závislosti musí být mezi tagy <head>, zbylé výpisy kdekoliv mezi tagy <body>.

Ukázka procedury PL/SQL určené k získávání dat pro potřeby vizualizace

Autor se rozhodl dle návrhu zanechat část logiky v PL\SQL proceduře. Tato procedura obsahuje přes 800 řádků kódu. Zde bude ukázán a vysvětlen pouze její fragment, spolu s možnostmi pro zobrazování dat. Celá procedura je složená z podmínek. Podmínka je pravdivá, pokud je parametr type ve vstupním parametru typu JSON Objekt rovný pravé straně výrazu. Na ukázce fragmentu je podmínka pro type roven „new_models_models_of_brand“

```
IF v_type = 'new_models_models_of_brand' THEN
SET v_lim :=
JSON_UNQUOTE(JSON_EXTRACT(p_json, '$.lim'));
SET v_model =
JSON_UNQUOTE(JSON_EXTRACT(p_json, '$.model'));
SELECT st_brands_name INTO v_brand FROM st_models JOIN st_brands ON st_brands.id =
st_models_brands_id WHERE st_models_name = v_model;
SELECT st_models_name AS 'lbl', SUM(st_model_statistics.st_model_statistics_count) AS 's' FROM
st_model_statistics JOIN st_models_groups ON st_model_statistics_models_groups_id =
st_models_groups_id
JOIN st_models ON st_models.id = st_models_groups_models_id
JOIN st_brands ON st_brands.id = st_models.st_models_brands_id
WHERE st_brands_name = v_brand AND
MAKEDATE(st_model_statistics_date_year,st_model_statistics_date_month) BETWEEN
MAKEDATE(v_year_from, v_month_from) AND MAKEDATE(v_year_to, v_month_to)
GROUP BY st_models_name
ORDER BY s DESC LIMIT v_lim;
END IF;
```

Fragment kódu přiřadí do proměnné v_lim hodnotu z JSON objektu lim, do proměnné v_model hodnotu z JSON objektu model. V další fázi podmínky procedury je z databáze vybrán záznam z tabulky st_models s připojenou tabulkou st_brands, díky čemuž se přiřadí do proměnné v_brand název značky pro model. Následuje SQL dotaz vracející celkové množství prodaných modelů automobilů pro danou značku z v_brand, odpovídajícím proměnným v_year_from, v_month_from, v_year_to a v_month_to limitován hodnotou z proměnné v_lim.

V zdrojovém kódu níže je ukázáno volání procedury pro dotazování dat. Z ukázky je také patrné, že procedura přijímá zřetěžený JSON objekt.

```
CALL selectNewBrands(
 '{"type":"new_brands_of_brand_fuels","month_from":"1","year_from":"2020","month_to":"12","year_to":"2020","lim":"5","brand":"Audi"}' )
```

Následuje seznam všech možných hodnot v hodnotě type JSON objektu.

- new_brands – Celkově značky.
- new_models_of_brand – Celkově všechny modely jedné značky.
- new_models – Celkově všechny modely.
- new_groups – Celkově všechny skupiny.
- new_fuels – Celkově všechna paliva.
- old_brands – Celkově všechny ojeté vozy dle značky.
- new_models_of_model_fuels – Celkově všechny paliva pro model.
- new_model_categories / new_model_categories_n_sum – Celkově všechny kategorie pro model.
- new_models_of_model_category_n_concurrency – Celkově okolí modelu pro jeho nejpočetnější kategorii.
- new_models_of_model_category_n_concurrency_per_month - Okolí modelu pro jeho nejpočetnější kategorii po měsících.
- new_models_of_model_n_concurrency – Celkově model a jeho největší konkurenti (dle nejpočetnější kategorie), sečteny všechny kategorie modelu.
- $\frac{\text{new_models_per_month}}{\text{new_models_of_model_n_concurrency_per_month}} / \frac{\text{new_brands_models_of_brand_per_month}}{\text{new_brands_models_of_brand_per_month_all}}$ – Jeden model po měsících.
- new_brands_models_of_brand_per_month_all – Všechny modely značky po měsících.
- new_brands_models_of_brand_per_year_all – Všechny modely značky po rocích.
- new_models_with_concurrency_per_month_all – Všichni konkurenti modelu po měsících.
- new_models_with_concurrency_per_year_all – Všichni konkurenti modelu po rocích.
- new_groups_per_month – Jedna skupina po měsících.
- new_groups_brands_per_month – Jedna skupina značky po měsících.
- new_groups_brands_total_score – Celkově značky ve skupině v porovnání s konkurenty.
- new_groups_fuels_per_month – Paliva v kategorii po měsících.
- new_groups_fuels_total_score – Paliva v kategorii celkově.
- new_groups_models_per_month – Modely v kategorii po měsících.
- new_groups_models_total_score – Celkově modely v kategorii.
- new_fuels_per_month – Paliva po měsících.
- $\frac{\text{new_brands_per_month}}{\text{new_brands_of_brand_n_concurrency_per_month}}$ – Jedna značka po měsících
- new_models_of_model_fuels_per_month – Jeden typ paliva pro model po měsících.
- new_models_of_model_fuels_per_month_all – Všechny paliva pro model po měsících.
- new_models_of_model_fuels_per_year_all – Všechny paliva pro model po rocích.
- new_models_total_score – Celkově modely v porovnání s konkurenty.
- new_brands_total_score – Celkově značky v porovnání s konkurenty.
- new_brands_of_brand_n_concurrency – Celkově konkurenti značky dle nejprodávanější kategorie.
- new_brands_with_concurrency_per_month_all – Konkurenti značky po měsících dle nejprodávanější kategorie
- new_brands_with_concurrency_per_year_all – Konkurenti značky po rocích dle nejprodávanější kategorie.

- new_brands_of_brand_fuels – Celkově paliva jedné značky.
- new_brands_models_of_brand – Celkově modely jedné značky.
- new_models_models_of_brand – Celkově modely jedné značky.
- new_brands_groups_of_brand – Celkově kategorie jedné značky.
- new_brands_of_brand_fuels_per_month – Jedno palivo značky po měsících.
- new_brands_of_brand_fuels_per_month_all – Všechny paliva značky po měsících.
- new_brands_of_brand_fuels_per_year_all – Všechny paliva značky po rocích.
- new_brands_groups_of_brand_per_month – Jedna kategorie značky po měsících.
- new_brands_groups_of_brand_per_month_all – Všechny kategorie značky po měsících.
- new_brands_groups_of_brand_per_year_all – Všechny kategorie značky po rocích.
- new_brands_colors – Všechny barvy značky.

Některé SQL dotazy v podmínkách jsou téměř duplicitní, liší se pouze v očekávaných vstupních parametrech – sestavení podmínek dotazu.

8 Testování

V této kapitole je popsáno testování komponenty. Komponenta byla vyvíjena déle než rok. Z hlediska zobrazování grafů a tabulek nebyla testována více, než při implementaci vykreslování grafů.

V průběhu vývoje byl však import statistik z portálu SDA-CIA několikrát ověřen. Bylo ověřeno nahrání všech tabulek od počátku, až po tabulky, které vycházely pravidelně každý měsíc. Během vývoje došlo ke změnám v tabulkách (přibyl nový model, značka, byla odstraněna záložka) a program přesto správně vložil data do databáze a upozornil na to uživatele. Během vývoje byly zjištěny chyby v datech, které také vyplývají z kapitoly [2.1] a musely být upraveny přímo v databázi. Tyto chyby, jak bylo zmíněno v analýze zdrojů, vznikly nejspíše lidskou chybou.

Mapování modelů z EuroNCAP na databázi existujících modelů z tabulek portálu SDA-CIA proběhl buď správně, či se nevyplnil. Toto chování je dle autora této práce správné. Dodatečné mapování probíhá přímo úpravou záznamů v databázi. Je na správci komponenty – databáze, aby určil dodatečné mapování především kvůli nutnosti rozhodnutí o relevanci podobnosti modelů.

9 Návrh budoucího rozšíření a inovace

V této kapitole jsou zmíněny některé body implementace, které lze vylepšit, případně do řešení přidat.

API

Ačkoliv při vývoji je počítáno s možností implementace API, implementována není. Komponenta – především třída pro komunikaci s databází však podporuje transformaci dat do přehlednější struktury. Vzhledem k faktu, že tato struktura je pole, které dokáže programovací jazyk serializovat je i toto řešení možné, není však implementován a otestován přístupový bod.

Samostatná šablona pro Javascript

PHP kód starající se o vykreslení JavaScriptu pracuje se spojováním řetězců. Toto řešení není přehledné a mnohem vhodnější by bylo rozdělit kód do externího souboru, který by sloužil jako šablona. Do této šablony by byly předány proměnné, které jsou nutné pro její správné vykreslení spolu s anotací pro případné napovídání v moderních IDE. Výsledný buffer s řetězcem by se před čtením šablony (externího souboru) nastavil pomocí `ob_start` a poté získal pomocí metody `ob_get_clean()`. Výsledek by byl mnohem přehlednější.

Informace o nově přidaných objektech

Samotný výpis nově přidaných modelů, značek, tříd, či paliv při importu tabulek z portálu SDA-CIA je pouze vypsán v tabulce. Při plném zautomatizování procesu pomocí cron úlohy byt tak uživatel neměl možnost, jak danou tabulku získat. Řešením by bylo např. odesláním tabulky emailem, zápisem do externího souboru atp.

Informace o chybách a jejich řešení v případě importu

Program jak při importu, tak při vykreslování grafů informuje o chybách opět výpisem chyby na stránku. V případě importu v tom není velký problém, pokud nenastane závažná chyba importu dat z tabulek, při které je proces nekompletní. Prostým řešením statistické záznamy pro daný měsíc a rok smazat a spustit program znova. Tento proces by šel řešit určitě lépe – transakce v databázi, automatické smazání a znovu nahrání atp. Při vykreslování grafu komponenty jsou ošetřeny vstupy, v případě chyby je však výpis velmi jednoduchý s okamžitým ukončením programu.

Vzhled grafů

Vzhled komponent by se dal více zparametrizovat. Knihovna použitá pro vykreslování grafů sama o sobě disponuje velkým množstvím jednoduše upravitelných parametrů pro vzhled. Jejich napojení by tak nemuselo být složité.

Interaktivní tabulky

Interaktivní tabulky jsou řešeny pouze formou odkazování na příslušný graf a efekty přejetí myši pomocí CSS. Autor této práce nenašel vhodnou inspiraci pro lepší chování. Tabulky mohly alespoň podporovat více verzí zobrazování dat, nejen po měsících.

10 Závěr

Práce po praktické části přináší nové řešení vizualizace a zpracování prodejů automobilů v ČR. Implementace automatizovaného nástroje a znovupoužitelné komponenty byly dosaženy s pár nedostatky v integraci řešení do externího systému – databáze. Ve srovnání s ostatními podobnými nástroji je přesto výsledná implementace unikátní. Výsledek je vizuálně přijatelný obzvláště pak v porovnání s ostatními statistikami.

Teoretická část nepřináší nové poznatky. Práce pouze shrnuje aktuální řešení, seznamuje s ním čtenáře a v případě knihoven pro vizualizaci dat podrobně mapuje aktuálně tři běžně používané knihovny. Práce ukazuje jejich praktické použití. Shrnutí ve výsledné tabulce knihoven porovnává jejich přednosti. Analýza a návrh podrobně vysvětlují proč a jak je výsledný program implementován.

Pro autora je práce novou zkušeností díky svému rozsahu a programovacím jazykům. Autor se díky práci naučil programovací jazyk PHP, JavaScript spolu s použitím frameworku Bootstrap 4 a rozšiřujících knihoven pro zhotovení výsledné implementace. Ačkoliv některé části práce považuje sám autor za slabší, kvůli způsobu implementace, výsledný program splnil jeho očekávání.

Použitá literatura

- [1] SDA. SDA [online]. Copyright © SDA [cit. 20.04.2021]. Dostupné z: <http://portal.sda-cia.cz/clanek.php?id=74>
- [2] Marklines. marklines.com [online]. Copyright © SDA [cit. 20.04.2021]. Dostupné z: http://www.marklines.com/portal_top_en.html
- [3] Best-selling-cars. Best-selling-cars.com [online]. Copyright © SDA [cit. 20.04.2021]. Dostupné z: <http://best-selling-cars.com>
- [4] Using Google Charts | Google Developers. Google Developers [online]. Dostupné z: <https://developers.google.com/chart/interactive/docs>
- [5] 20 best JavaScript charting libraries. TNW | The heart of tech [online]. Copyright © 2006 [cit. 20.04.2021]. Dostupné z: <https://thenextweb.com/news/20-best-javascript-chart-libraries>
- [6] Chartist - Simple responsive charts. Chartist.js· GitHub Pages [online]. Dostupné z: <https://gionkunz.github.io/chartist-js/>
- [7] Chart.js | Open source HTML5 Charts for your website. Chart.js | Open source HTML5 Charts for your website [online]. Dostupné z: <https://www.chartjs.org/>
- [8] Usage Statistics and Market Share of PHP Version 7 for Websites, April 2021. W3Techs - extensive and reliable web technology surveys [online]. Copyright © 2009 [cit. 20.04.2021]. Dostupné z: <https://w3techs.com/technologies/details/pl-php/7>
- [9] Welcome to PhpSpreadsheet's documentation - PhpSpreadsheet Documentation. [online]. Dostupné z: <https://phpspreadsheet.readthedocs.io/en/latest/>
- [10] PHP Simple HTML DOM Parser. PHP Simple HTML DOM Parser [online]. Dostupné z: <https://simplehtmldom.sourceforge.io/>
- [11] CSS: Cascading Style Sheets | MDN. [online]. Copyright © 2005 [cit. 20.04.2021]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [12] JavaScript – Wikipedie. [online]. [cit. 20.04.2021]. Dostupné z: <https://cs.wikipedia.org/wiki/JavaScript>
- [13] Tvorba infografiky | FF - KISK: Kurz práce s informacemi. Katedra informačních studií a knihovnictví [online]. Copyright © 2018 Masarykova univerzita [cit. 20.04.2021]. Dostupné z: <https://kisk.phil.muni.cz/media/3089574/kisk.phil.muni.cz/kpi/temata/vizualizace/tvorba-infografiky.html>
- [14] Informační systém [online]. Copyright © [cit. 20.04.2021]. Dostupné z: <https://is.muni.cz/th/vn8jk/tvorba-efektivnich-grafu.pdf>
- [15] Procesní analýza (Process analysis) - ManagementMania.com. [online]. Copyright © 2011 [cit. 20.04.2021]. Dostupné z: <https://managementmania.com/cs/analyza-procesu-procesni-analyza>
- [16] Unified Modeling Language – Wikipedie. [online]. Dostupné z: https://cs.wikipedia.org/wiki/Unified_Modeling_Language
- [17] Introducing Types of UML Diagrams | Lucidchart Blog. Online Diagram Software & Visual Solution | Lucidchart [online]. Copyright © 2021 Lucid Software Inc. [cit. 20.04.2021]. Dostupné z: <https://www.lucidchart.com/blog/types-of-UML-diagrams>

- [18] What Exactly Is Wireframing? A Guide To Wireframes. Become a designer, developer or data analyst from scratch [online]. Dostupné z: <https://careerfoundry.com/en/blog/ux-design/what-is-a-wireframe-guide/>
- [19] Jak by měla vypadat úvodní stránka webu? Víme, jaké prvky by měla obsahovat | Marketup. [online]. Copyright © Marketup s.r.o. 2020 [cit. 20.04.2021]. Dostupné z: <https://www.marketup.cz/cs/blog/jak-by-mela-vypadat-uvodni-stranka-webu-vime-jake-prvky-by-mela-obsahovat>
- [20] Co to je infografika a v čem vám pomůže | ČeskéInfografiky.cz. ČeskéInfografiky.cz [online]. Copyright © 2021 [cit. 20.04.2021]. Dostupné z: <https://ceskeinfografiky.cz/co-je-infografika/>
- [21] Webový server - - HTML5. HTML5 - [online]. Copyright © 1997 [cit. 20.04.2021]. Dostupné z: <http://htmlguru.cz/vystaveni-webovy-server.html>
- [22] PHP: MySQL Database. W3Schools Online Web Tutorials [online]. Dostupné z: https://www.w3schools.com/php/php_mysql_intro.asp
- [23] HTML příručka, přehled HTML tagů. Jak psát web, návod na html stránky [online]. Dostupné z: <https://www.jakpsatweb.cz/html/>
- [24] About the Apache HTTP Server Project - The Apache HTTP Server Project. Welcome! - The Apache HTTP Server Project [online]. Copyright © 1997 [cit. 20.04.2021]. Dostupné z: https://httpd.apache.org/ABOUT_APACHE.html

Seznam příloh

Příloha A: Záložky vhodné ke zpracování..... I.

Příloha B: Odpověď HTTP GET požadavku z APU EuroNCAP II.

Příloha A: *Záložky vhodné ke zpracování*

Data značek:

- OA za měsíc
- OA za měsíc Jiné značky
- LUV za měsíc
- LUV za měsíc Jiné značky
- Ojetiny OA za měsíc
- Ojetiny OA za měsíc jiné

Data modelů :

- OA Paliva za měsíc Mini
- OA Paliva za měsíc Malé
- OA Paliva za měsíc Nižší střední
- OA Paliva za měsíc Střední
- OA Paliva za měsíc Vyšší střední
- OA Paliva za měsíc Luxusní
- OA Paliva za měsíc Sportovní
- OA Paliva za měsíc SUV B
- OA Paliva za měsíc SUV C
- OA Paliva za měsíc SUV D
- OA Paliva za měsíc SUV E
- OA Paliva za měsíc MPV
- OA Paliva za měsíc MPV B
- OA Paliva za měsíc MPV C
- OA Paliva za měsíc MPV E
- OA Paliva za měsíc Terénní
- Hybridy za měsíc
- LUV Paliva za měsíc Do 2 0 t
- LUV Paliva za měsíc Do 3 0 t
- LUV Paliva za měsíc Do 3 5 t

Pro zjištění použitých tříd:

- OA Podíl tříd

Pro zjištění použitých paliv:

- OA Paliva za měsíc Mini
- Hybridy za měsíc

Data barev:

- Barvy OA Měsíc
- Barvy OA Měsíc Ostatní

Data ojetin:

- Ojetiny OA za měsíc
- Ojetiny OA za měsíc jiné

Příloha B: *Odpověď HTTP GET požadavku z API EuroNCAP*

```
{
  "AssessmentSearchResults":[
    {
      "ProtocolId":24370,
      "ProtocolYear":2016,
      "Assessments":[
        {
          "Stars":5,
          "OverallRating":81,
          "ChildOccupantRating":81,
          "AdultOccupantRating":98,
          "PedestrianRating":69,
          "SafeAssistRating":60,
          "StarsWithOverallRating":581,
          "DriverAssistanceTechnologyIds":"0",
          "DriverAssistanceTechnologies":"0",
          "AdvancedRewardTechnologyIds":"0",
          "AdvancedRewardTechnologies":"0",
          "AdvancedRewardTechnologiesCount":0,
          "IsHybrid":false,
          "FrontCarImage":"https://cdn.euroncap.com/media/23273/g_34frontwhite.jpg",
          "CrashCarImage":"https://cdn.euroncap.com/media/23261/alfa_giulia_2016_odb-image-gallery.jpg",
          "FullSafetyRatingAssessmentId":0,
          "StandardSafetyRatingAssessmentId":0,
          "StandardSafetyPack":true,
          "FullSafetyPack":false,
          "Xml":null,
          "XmlUploaded":false,
          "HideFromHomepageCarousel":false,
          "BestInClassCarClassEN":null,
          "BestInClassCarClassDE":null,
          "BestInClassCarClassNL":null,
          "BestInClassCarClassES":null,
          "BestInClassCarClassFR":null,
          "BestInClassCarClassIT":null,
          "BestInClassCarClassRU":null,
          "BestInClassCarClassSV":null,
          "BestInClassCarClassTR":null,
          "BestInClassCarClassZH":null,
          "BestInClassCarClassId":0,
          "Id":24849,
          "Title":"Alfa Romeo Giulia",
```

```

{Name": "Alfa Romeo Giulia",
Make": "Alfa Romeo",
Model": "Giulia",
Content": "test",
Url": "/results/alfa-romeo/giulia/24849",
Year": 2016,
ClassEN": "Large Family Car",
ClassDE": "Mittelklasse",
ClassNL": "Grote gezinsauto",
ClassES": "Familiar grande",
ClassFR": "Grande familiale",
ClassIT": "Familiare grande",
ClassRU": "Большой семейный автомобиль",
ClassSV": "Stor familj",
ClassTR": "Büyük Aile",
ClassZH": "大型家用车",
ClassId": 1201,
MakeImageLarge": "https://cdn.euroncap.com/media/4457/alfa-romeo-80x53.jpg",
MakeImageSmall": "https://cdn.euroncap.com/media/4458/alfa-romeo-62x42.jpg",
ProtocolId": 24370,
MakeId": 7276,
ModelId": 24848,
IsPublished": true,
CreateDate": "\Date(1465207614000)\",
ReleaseDate": "\Date(1466553600000)\",
Images": "https://cdn.euroncap.com/media/23260/alfa_giulia_2016_odt1-image-gallery.jpg,https://cdn.euroncap.com/media/23259/alfa_giulia_2016_fw-image-gallery.jpg,https://cdn.euroncap.com/media/23264/alfa_giulia_2016_side1-image-gallery.jpg,https://cdn.euroncap.com/media/23263/alfa_giulia_2016_pole-image-gallery.jpg,https://cdn.euroncap.com/media/23262/alfa_giulia_2016_pole1-image-gallery.jpg",
YoutubeVideos": "https://youtu.be/7YkddjbcQew",
YoukuVideos": "http://player.youku.com/player.php/sid/XMTc0MDI2MzYyOA==/v.swf",
IsPreProtocol": false,
IsFleet": true,
SortOrder": 8,
LifecycleStatus": 25,
FirstPublished": "\Date(1466553600000)\",
}
]
},
"PreAssessmentSearchResults": [
]
}

```
